Digital Stack Photography and Its Applications

by

Jun Hu

Department of Computer Science
Duke University

Date: _____

Approved:

_____
Xiaobai Sun, Supervisor

_____
Michael Gehm

_____
Mauro Maggioni

_____
Nikos Pitsianis

_____
John Reif

_____
Guillermo Sapiro

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Computer Science
in the Graduate School of Duke University
2014

Abstract

Digital Stack Photography and Its Applications

by

Jun Hu

Department of Computer Science
Duke University

Date: _____

Approved:

_____
Xiaobai Sun, Supervisor

_____
Michael Gehm

_____
Mauro Maggioni

_____
Nikos Pitsianis

_____
John Reif

_____
Guillermo Sapiro

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science in the Graduate School of Duke University
2014

Abstract

This work centers on digital stack photography and its applications. A stack of images refer, in a broader sense, to an ensemble of associated images taken with variation in one or more than one various values in one or more parameters in system configuration or setting. An image stack captures and contains potentially more information than any of the constituent images. Digital stack photography (DST) techniques explore the rich information to render a synthesized image that oversteps the limitation in a digital camera's capabilities. This work considers in particular two basic DST problems, which had been challenging, and their applications. One is high-dynamic-range (HDR) imaging of non-stationary dynamic scenes, in which the stacked images vary in exposure conditions. The other is large scale panorama composition from multiple images. In this case, the image components are related to each other by the spatial relation among the subdomains of the same scene they covered and captured jointly. We consider the non-conventional, practical and challenge situations where the spatial overlap among the sub-images is sparse (S), irregular in geometry and imprecise from the designed geometry (I), and the captured data over the overlap zones are noisy (N) or lack of features. We refer to these conditions simply as the S.I.N. conditions.

There are common challenging issues with both problems. For example, both faced the dominant problem with *image alignment* for seamless and artifact-free image composition. Our solutions to the common problems are manifested differently

in each of the particular problems, as a result of adaption to the specific properties in each type of image ensembles. For the exposure stack, existing alignment approaches struggled to overcome three main challenges: inconsistency in brightness, large displacement in dynamic scene and pixel saturation. We exploit solutions in the following three aspects. In the first, we introduce a model that addresses and admits changes in both geometric configurations and optical conditions, while following the traditional optical flow description. Previous models treated these two types of changes one or the other, namely, with mutual exclusions. Next, we extend the pixel-based optical flow model to a patch-based model. There are two-fold advantages. A patch has texture and local content that individual pixels fail to present. It also renders opportunities for faster processing, such as via two-scale or multiple-scale processing. The extended model is then solved efficiently with an EM-like algorithm, which is reliable in the presence of large displacement. Thirdly, we present a generative model for reducing or eliminating typical artifacts as a side effect of an inadequate alignment for clipped pixels. A patch-based texture synthesis is combined with the patch-based alignment to achieve an artifact free result.

For large-scale panorama composition under the S.I.N. conditions, we have developed an effective solution scheme that significantly reduces both processing time and artifacts. Previously existing approaches can be roughly categorized as either geometry-based composition or feature based composition. In the former approach, one relies on precise knowledge of the system geometry, by design and/or calibration. It works well with a far-away scene, in which case there is only limited variation in projective geometry among the sub-images. However, the system geometry is not invariant to physical conditions such as thermal variation, stress variation and etc.. The composition with this approach is typically done in the spatial space. The other approach is more robust to geometric and optical conditions. It works surprisingly well with feature-rich and stationary scenes, not well with the absence of recognizable

features. The composition based on feature matching is typically done in the spatial gradient domain. In short, both approaches are challenged by the S.I.N. conditions. With certain snapshot data sets obtained and contributed by Brady *et al*, these methods either fail in composition or render images with visually disturbing artifacts. To overcome the S.I.N. conditions, we have reconciled these two approaches and made successful and complementary use of both priori and approximate information about geometric system configuration and the feature information from the image data. We also designed and developed a software architecture with careful extraction of primitive function modules that can be efficiently implemented and executed in parallel. In addition to a much faster processing speed, the resulting images are clear and sharper at the overlapping zones, without typical ghosting artifacts.

To my famliy

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

Ω	A 2D image domain.
$\mathbf{x}$	A pixel in Ω.
p	Patch height and width.
$\mathbf{P}_{\mathbf{x}}$	A $p \times p$ patch centered at pixel $\mathbf{x}$.
$\mathbf{u}$	A flow/vector field.
τ	A brightness transfer function.
f	A camera response function.
R	A reference or target image.
R^{τ}	A intensity mapped reference image under τ
S	A source image.
$S_{\mathbf{u}}$	A warped source image under flow field $\mathbf{u}$.
L	A latent image.
I	An image.
$\mathbf{H}$	A homography transformation matrix.
$\mathbf{x} \leftrightarrow \mathbf{x}'$	A pair of feature match.

Acknowledgements

First of all I would like to express my sincere thanks to my advisor, Professor Xiaobai Sun, for her intense support, guidance, and advice for my study and life in U.S.A. It is such a treasurable and unique experience for me to be able to pursue my Ph.D under the guidance of Xiaobai. In addition to helping me on the technical aspects, she encouraged me to find and pursue what interested, inspired and motived me the most, which led to this dissertation topic and work. I learned from her how to investigate a research topic in depth and in connection to the others and how to develop myself in a new field.

My deep gratitude also goes to Professor Michael Gehm., Professor Mauro Maggioni, Professor Nikos Pitsiani, Professor John Reif, Prof. Guillermo Sapiro, Professor Nikos Pitsiani and Professor Rebecca Willett who is at university of Wisconsin Madison now, for their advices and support while serving on the committees for my initial research project, master thesis project, preliminary exam and the dissertation defense. In addition, Professor Nikos Pitsianis has been a wonderful collaborator and mentor through my graduate study at Duke University. Professor Michael Gehm. and his research collaborators have provided me valuable references, documents and data that I used in part in my research work. I have been inspired and influenced by Professor Guillermo Sapiro 's work, and I consider myself fortunate to have him serving on my dissertation committee.

I am very grateful to my mentor Dr. Orazio Gallo and manager Dr. Kari Pulli at

Nvidia Research where I had interned twice. They introduced me to the relatively new field, computational photography, and influenced me with great passion, knowledge and experience. As my mentor, Orazio also gave me many valuable suggestions and helps not only for my short-term internships but also for my long-career development. He taught me to persevere while facing difficulties and failures in research.

I thank my friend and research collaborator Alexandros Iliopoulos for his stimulating ideas, his strong problem-solving skills in our collaboration as well as his Greek humor. To support my full time job at Apple, Alexandros kindly took over my TA responsibility at the expense of his own research or relax time, which I don't know how to pay back. I also want to thank my other friends at Duke University – Ang Li, Yi Hong, Lijun Yao, Qiang Cao, Xin Wu, Xuanran Zong, Yu Chen and Yang Chen. They make my graduate life more enjoyable and memorable.

Finally and most importantly, I thank my family. Shanshan, my wife, is a constant inspiration for me since our college days. She inspires me more by getting her PhD in Mathematics ahead of me. She has given me unwavering support and inexpressible joy and happiness. My parents have loved me unconditionally, they have installed in me important life values. I thank my brother Li Yu for his support and encouragement in pursuing my dreams. Last, I have inexpressible joy and sense of duty for the arrival of my son Leo a month before my dissertation defense.

1

Introduction

The photograph isn't good enough. It's not real enough.

David Hockney, 1937 - Present

We are interested in digital stack photography (DSP) and its applications. A stack of images refer, in a broader sense, to an ensemble of associated images taken with variation in one or more than one parameters in system configuration or setting. In the study of digital stack photography, we are concerned with how to design, acquire and process a stack of images in order to lift or relieve certain limitations in single-image photography with the existing optical systems or cameras, and to render more realistic or desirables images of the objects or sceneries under various circumstances. The underlying principle is the same as that for computational tomography (CT). In theory and practice, the acquisition and composition enjoy certain freedom while subject to particular constraints and limitations in camera devices and photo taking environments. We describe briefly in this section the DST background,

advanced technologies and techniques, certain remaining limitations, and two basic DSP problems we are concerned with in this thesis.

1.1 Advances in Digital Photography

Digital stack photography (DST) techniques have advanced in multiple and integral aspects. In one main aspect, image stacks are designed to be able *capture* and contain potentially more information. Innovative digital photography devices and techniques have enabled practical stack image acquisition at fast speed. In another important computational processing techniques have been richly developed for image synthesis and composition with image stacks or ensembles.

As the name suggests, DST stems and evolves from photography. Photography offers an unique and powerful representation of reality. For many of us, photographs are a medium to record special moments, to preserve memories, to tell and share personal stories, or enhance news reporting with visual effects. Artists use photography to express their visions of the world, their feelings, ideas and thoughts. Scientists use photography to open new fields of exploration to widen or augment man's visual and intellectual horizons. For instance, physicists deploy high-speed photography to explore the behavior of granular materials and the physics of the nuclear bomb, biologists adopt time-lapse photography to record the plants growing and flowers opening, and astronomers use astrophotography to study celestial objects.

Much progress has been made to assist photographers in producing better photographs, especially, in natural but difficult circumstances. There are significant changes and advances in optical hardware and electronic hardware, from lenses to sensors. For instance, novel synthetic materials like fluorite crystals are used to make lens to reach a lower dispersion than glass. The size of photodiode in image sensor becomes smaller and allows to capture up to 16 thousands of pixels in one single shoot. More powerful microprocessors are integrated with individual sensors

to allow a faster image formation.

Nonetheless, existing digital camera systems are still subject to certain limitations and constraints. For example, the maximal aperture size , the minimal focal length of lens and the maximal resolution of imaging sensor are limited in a small and light camera system such as those digital cameras we carry in our pockets or mounted on our cell phones. These optical parameters affect the quality of the acquired images. The focal length of the lens determines the amount of the scene projected onto the image area, the size of the digit sensor determines the size of image area, and the resolution of imaging sensor is an important measure of how much details has been recorded to a certain level of contrast.

1.2 Computational Stack Image Processing

Computational stack image processing is enabled by the advances in hardware advances and extracts more information from the captured images. Software is used to assist, enhance or complement with the hardware at almost every stage in the photography process, from shutter setting, aperture and focus control in acquisition, noise reduction, while balancing and image compression in pre-processing, to cropping, sharpening, toning in post-processing.

While a single image generated by a common digital camera is limited in capturing information, a stack of photographs with different camera parameters can contain much more information as we desire. With the advances in photography devices and techniques, it is becoming possible and practical to acquire multiple images with one shot, extract information and reduce noise from the acquired ones, and compose better ones what are beyond the capabilities of conventional photography. An early attempt and application of digital stack photograph is the so called focal stacking (F.Ray, 2002). By this method, multiple images taken at different focus distances are intelligently merged into a single image with a greater depth of field.

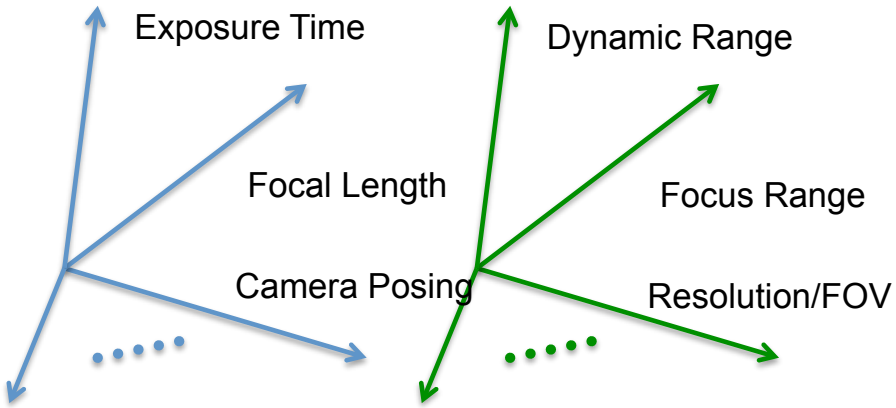


FIGURE 1.1

Another interesting work is flash/no-flash stack (Petschnigg et al., 2004), in which multiple images are captured at different flash and gain configurations, are combined to obtain acceptable results under a low-light environment. Nowadays, the shutter speed of digital camera is becoming increasingly fast and the size of image sensor tends to smaller. All these offer stack photography many opportunities to explore in the temporal and spatial dimension.

Computational stack image processing synthesizes the stack images, acquired with one or more camera parameters $\mathbf{X}$ varying, to improve one or more than one characteristics $\mathbf{Y}$. Typically, $\mathbf{X}$ parameters are exposure time, ISO gain, aperture and focus, and desirable $\mathbf{Y}$ characteristics are high dynamic range (HDR), high signal-to-noise ratio (SNR), high resolution and seamless panorama (See Fig 1.1). This dissertation considers in particular two basic DST problems, which had been challenging, and their applications. One is high-dynamic-range (HDR) imaging of non-stationary dynamic scenes, in which the stacked images vary in exposure. The other is large scale panorama composition from multiple images. In this case, the image components are related to each other by the spatial relation among the sub-domains of the same scene they covered and captured jointly. We consider the non-

conventional, practical and challenge situations where the spatial overlap among the sub-images is sparse (S), irregular in geometry and imprecise from the designed geometry (I), and the captured data over the overlap zones are noisy (N) or lack of features. We refer to these conditions simply as the S.I.N. conditions.

1.3 Dynamic Range

In photography, the dynamic range refers to the ratio between the maximum and minimum measurable light intensities (McHugh, 2005). In other words, it is the ratio in intensity between the between the lightest and darkest regions, and it is therefore also referred to as the contrast ratio. Note the keyword *measurable*. For a real world scene, the light intensity can be described in terms of strength of the incident and reflected light, which are typically measured in candelas per square meter (cd/m^2). Table 1.1 gives approximate light intensities for common incident light sources. The intensity of reflected light dramatically varies with the reflectance of subject surface and relative position with lighting source For a dim interior with view through window to bright sunlight, the light intensity range might be $10^{-2}cd/m^2$ to $10^5 cd/m^2$, which corresponds to a dynamic range of $10^7 : 1$ (Freeman, 2008).

Table 1.1: Approximate Light Intensity of Common Incident Lights

Subject	Approximate light intensity (cd/m^2)
Starlight	0.001
Moonlight	0.1
Indoor lighting	100
Indoor daylight	100
Cloudy day	2,000
Open Shade	10,000
Bright sunlight	100,000
Disc of the sun itself	100,000,000

The dynamic range is defined slightly different for light intensity measurement devices such as digital camera sensor. It refers to the practically capacity of recording and representing the variations or range in intensity (McHugh, 2005). For a digital image sensor, the dynamic range is the ratio of maximum number of photons a pixel sensor (also called photosite) could contain before gets saturated, to the minimum number of detectable photons. Nowadays, an affordable digital camera could have dynamic range up to $\approx 4,000:1$, some expensive camera can go up to $10^4:1 \sim 10^5:1$. In comparison to the dynamic range of real world scenes of common interests, the images captured by common digital cameras are of low dynamic range image (LDR).

One way to increase the dynamic range would be to increase the maximum capacity by building a large sensor that allows to receive a greater flux of photons. However, a larger sensor typically costs more and requires more power to support. The other way to extend the dynamic range would be to lower the minimum number of detectable photos, which depends on the sensor base noise level. The base noise level is affected by many factors, such as sensor temperature and photodiode leakage current, which remain difficult to control with existing manufacturing technologies and techniques.

With digital high dynamic range (HDR) imaging, one attempts to match up to the world scenes of high dynamic range in light intensity with limited capacities in photo capturing and recording. There are two basic ways to capture the full dynamic range (Reinhard et al., 2010). The first conceivable way is to design novel imaging system to shoot an HDR image directly. A promising direction is spatially varying exposure - a pixel pattern of different exposures (Nayar and Mitsunaga;, 2002). The dynamic range is expanded by 'intelligently' combine neighboring pixels with different exposures. An alternative solution is to mix different light sensitivity photodiodes in one sensor - one captures regular image, and the other captures the highlight details - both images are combined to a wide dynamic range. Instead of

packing more photodiodes, a more clever way is to use a semitransparent mirror to split the incoming lights to capture different exposures on different sensors.

An alternative way would be to shoot a series of LDR images with bracketed exposures. While each captured image is of low dynamic range, a single HDR image may be generated from the LDR images. This can be done with any camera will adjustable exposure setting. Because of its ease implementation, HDR imaging based on bracketed exposures has became standard feature in many commercially massively produced cameras. We consider in this thesis this type of HDR imaging, namely, *HDR reconstruction*.

FIGURE 1.2: An example of reconstructing HDR from LDRs. Left column: Three low dynamic. range images (LDRs) captured by a standard digital camera; Right : Tone mapped HDR image composed from the three LDR images to the left. Note that the dynamic range of a single shot is much lower than the dynamic range of scene. This results in the loss of details in bright (sky) or dark areas(tree). However, a HDR image can be well reconstructed by intelligently merging LDRs together to cover the contrast in both dark and bright areas.

1.4 Live Scenes

Despite its simplicity, existing HDR reconstruction approaches require multiple, perfectly aligned pictures taken at different exposure levels. Despite best efforts to keep the camera stable in capturing, movement from image to image is always unpreventable. Motion modeling is needed for motion compensation. In certain constrained situation, such as stationary scenes, the movement can be reasonably assumed to be a globally rigid transformation. Many existing feature-based alignment techniques can be used for motion compensation. For more practical situations, one take into account of non-rigid motion, which for example is commonly described in terms of *optical flow*, a displacement field, over each and every pixel, from the picture at one moment to that at the next moment (Szeliski, 2010). The underlying assumption is *brightness constancy*, namely, the pixel brightness is invariant to the change in its associated pixel location. In the exposure stack, it is intended to have the brightness change from one image to the next in order to capture the local contrast. HDR imaging of live scene with exposure image stack is therefore different in motion model and in motion compensation from the conventional approach.

Figure 1.3 shows a failure case of HDR for live scene using the conventional approach. Note the ghosting artifact due to people in motion and the parallax error (Window) due to camera motion.

FIGURE 1.3: An example of HDR for live scene; Left: Three low dynamic range (LDR) images ; Right: A tone mapped HDR image composed from the three LDR images using commercial software Photomatix (HDRsoft Ltd, 2003). The result is produced using software's default configuration.

1.5 Panoramic Views

A Panoramic view is any wide-angle view or representation of a physical space (London et al., 2007). In photography, angle of view describes the visible extent of the scene captured by the image sensor. Wide angle of views capture greater areas, small angles smaller areas. For standard digital camera, the angle of view completely depends on the effectively focal length of lens and the size of sensor. Figure 1.4 shows the relationship between the angle of view and the effectively focal length of lens for a standard digital camera with 35mm sensor size. The shorter the focal length, the wider the angle of view and the greater the are capture. Similar, the larger of sensor, the wider the angle of view (See Figure 1.5). Basically, there are two ways to capture a wide of angle image. The first way is to build a impractical large image sensor or a lens with effectively focal length less than 8mm. Many camera manufactories, like Nikon and Sigma, already produced 4.5mm fisheye lens, but one disadvantage of wide of angle lens is strong distortion caused by optical aberration.

Panoramic stitching allows the photographer a second way to create a wide angle of view image using a non wide angle view digital camera and lens. The basic idea is to shoot a series of images by rotating the camera about the optical center of its lens. While each image covers a limited angle of view, a wider angle of view image can be generated by stitching the images across different angles. Panoramic stitching allows to encompass a very wide angle of view, up to 360 degree panorama view and has much less distortion than fishy lens. However, achieving a seamless result is more complicated than just aligning photographs. The stitched panorama typically has drastic range of illumination across all image angles, an improperly blending may generate visual artifact at seam. Besides, the camera is required to rotates about the optical center of its lens, thereby maintaining the same point of perspective of all captured images; If the camera does not rotate about its optical

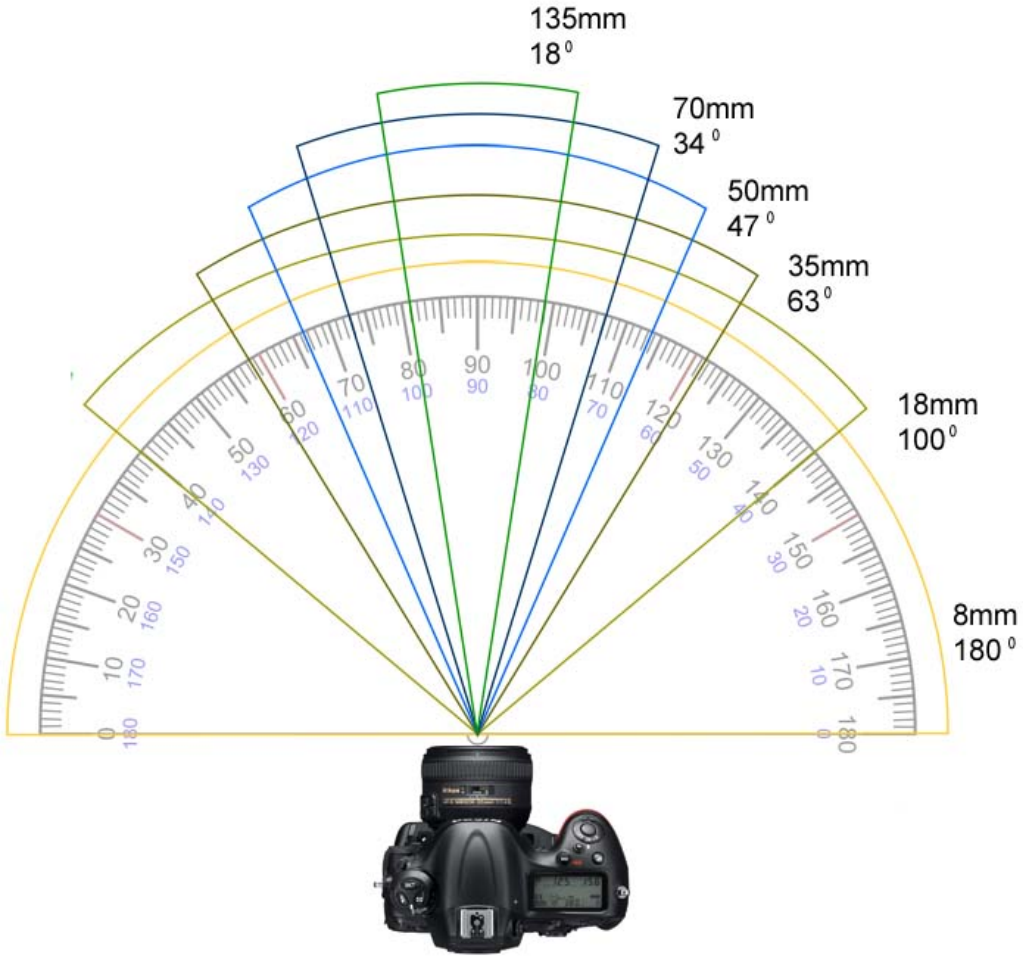


FIGURE 1.4: Angle of view by focal length for full-frame sensor (Free Photography Tutorials, lessons, tips, tricks DSLR Camera, 2014).

center, misalignment called parallax error would be introduced. In particular, the scenario with foreground subject is very sensitive to parallax error.

In recent years, as digital cameras becomes cheap, a popular trend in the field of panoramic stitching is to use an array of camera to capture images across all angles simultaneously, rather than rotate one camera in sequence. Basically, there are two apparent advantages of camera array. First, few artifacts would be introduced in non-stationary scene because all images are captured simultaneously. Second, if the optical center of all cameras are aligned to the same point, the parallax error can be

Image size (millimeters)				Angle of View (degree)		
Image format	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
35mm Still	36.0	24.0	43.3	90.0°	67.4°	100.5°
35mm Movie	22.1	16.0	27.3	63.0°	48.0°	74.3°
Super 16mm Movie	12.5	7.4	14.6	38.4°	23.3°	44.0°
2/3" 16:9 Wide screen	9.6	5.4	11.0	29.9°	17.1°	34.0°
2/3" 4:3 Cross Over	7.2	5.4	9.0	22.6°	17.1°	28.1°
2/3" 4:3 Standard	8.8	6.6	11.0	27.5°	20.8°	34.0°
1/2" 16:9 Wide Screen	6.97	3.92	8.0	21.9°	12.4°	25.1°
1/2" 4:3 Standard	6.4	4.8	8.0	20.2°	15.2°	25.1°
1/3" 16:9 Wide Screen	5.23	2.94	6.0	16.5°	9.3°	18.9°
1/3" 4:3 Standard	4.8	3.6	6.0	15.2°	11.4°	18.9°

FIGURE 1.5: Angle of view by sensor size for an effective focal length 18 mm (Cannon Inc., 2014).

resolved in theory. However, existing camera array system is designed to maximize the composited angle of view (or field of view) and gives serval challenges: (i) The amount of overlap between adjacent cameras is scare and has roughly less than 10% of the adjacent images. (ii) The cameras are packed on the dome-like mount, so the geometric distribution are highly irregular across adjacent cameras. (iii) Image data in regions of overlap are highly noisy, due to adverse vignetting and stray light effects. All these conditions really challenge existing stitching algorithms.

1.6 Dissertation Organization

In Chapter 2, we review the state of the art in HDR reconstruction. We describe three different methods for aligning exposure stack for dynamic scenes. In our first method, we show how conventional optical flow can be extended to deal with *brightness in-*

constancy cases. The next method we describe aims at handling large displacement. We replace the pixel-based optical flow with the patch-based model, and adopt a randomized algorithm to solve the new model in an efficient and reliable way even in the presence of large displacement. The last method we describe addresses the question of how to deal with clipped pixels in the reference image. We show that a reliable alignment for clipped pixels is difficult but the aligned content can be well synthesized under some heuristic guides.

In Chapter 3, we give a brief literature review on panoramic stitching techniques. We are interested in stitching images acquired by array cameras system, such as AWARE-2 (Golish et al., 2012), in which adjacent micro-cameras could fire simultaneously and are free of ghosting caused by moving object but have sparse, geometrically irregular and noisy (S.I.N) overlap. We propose a processing pipeline to deal with S.I.N challenge using the intrinsic properties of array camera system. We describe a placement geometric guided RANSAC to improve pairwise alignment for images with sparse overlap. Once the pairwise alignment has been obtained, we refine all camera parameters together through *global bundle adjustment*. Since global bundle adjustment error is non-invariant to projective camera matrix, we describe a global bundle adjustment by minimizing the measurement errors in the final mosaicing space. In the last stage, we show that a virtually ghosting-free final composition can be produced by merging in the image gradient domain.

In Chapter 4, we summarize findings of this dissertation and describe the limitations of our approaches. We also discuss directions that may be worth investigating in the future.

1.7 Disseminated Collaboration work

Most of the concrete contents in this dissertation has been disseminated in the following papers[1]:

- J. Hu, O. Gallo and K. Pulli, "Exposure Stacks for Live Scenes with Hand-held Cameras", European Conference on Computer Vision (ECCV), Florence, Italy, 2012

- J. Hu, O. Gallo, K. Pulli and X. Sun, "HDR Deghosting: How to deal with Saturation?", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, Oregon, 2013

- A. Iliopoulos, J. Hu, N. Pitsianis, X. Sun, M. Gehm, and D. Brady. "Big Snapshot Stitching with Scarce Overlap", IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, 2013

The dissertation is intended to address and emphasize on certain common and fundamental issues, in terms of digital stack photograph in a broader sense, among the specific and seemingly different synthesis problems.

[1] Joint work with Prof. Xiaobai Sun, Dr. Kari Pulli, Dr. Orazio Gallo, Prof. Nikos Pitsianis and Alexandros Stavros Iliopoulos

2

HDR Imaging of Live Scenes

Photograph: a picture painted by the sun without instruction in art.

Ambrose Bierce, 1842 - 1914

High-dynamic-range imaging (HDR) is one of early applications of stack photography. Gustave Le Gray combined two negatives, one for the sky and a longer exposure for the sea, to create a single image in 1850. In the mid nineteen century, Charles Wyckoff developed the HDR imaging we use today by combining differently exposed film layers into a single image. His famous, detailed photographs of nuclear explosions were featured on the front cover of Life magazine.

Why we need HDR ? A real-world scene varies over a wide range of brightness and might has a dynamic range up to 10,000,000:1, but a digital camera typically use 8 to 14 bits of brightness to encode the dynamic range at each pixel. This results in the loss of details in bright or dark areas. High-dynamic-range (HDR) imaging compensates for this loss by capturing a stack of differently exposed low-dynamic-

range pictures (LDR) of the same scene and intelligently merging them together to produce a picture that is representative in both dark and bright areas.

2.1 Challenges

While HDR is seemingly straightforward and sounds like a magic bullet for photography, performing it without visible artifacts requires considerable attention to detail. In particular, alignment of LDR is one of the major obstacles of HDR. Since the merging process assumes that the pixels of the different images are perfectly aligned, any motion — either due to the motion of the camera or anything moving in the scene, such as a blowing branch or a walking person — will cause *ghosting* artifacts (if the motion is large) or *blurring* artifacts (if the motion is small).

Several methods have been proposed that can deal with camera motion (Ward, 2003; Tomaszewska and Mantiuk, 2007) or dynamic scenes (Khan et al., 2006; Jacobs et al., 2008; Gallo et al., 2009; Zhang and Cham, 2010) often at the cost of discarding some of the information; but they produce sub-optimal results when both sources of artifacts are present. Our objective is to preserve most of the available information from the different shots: instead of disregarding misaligned pixels, our approach warps and modifies the content of each image in the stack to better align it with an image of the stack that we select as a reference.

The alignment of exposure stacks is still a challenging research topic. The underlying assumption behind most of conventional alignment techniques is *brightness constancy* — a pixel retains its brightness as it flows from image to image. Nonetheless, this assumption is impossible to meet for exposure stacks, because the pixel brightness is intend to change in order to effectively acquire the scene irradiance. An alternate approach is to match robust features (such as SIFT and BRIEF) between images in the exposure stack and estimate a global transformation by fitting the matched features. Despite its robust to brightness change, the approach only

accounts for *homography* transformation and is well known to fail for moving objects or camera motions that are more complex than a pure rotation.

To address this problem, we demonstrate three approaches: First, we adopt and modify the energy-based optical-flow approach to cope with the brightness changes. Then, we extend this energy-based approach to deal with large scale, non-rigid displacement, which is usually a unavoidable problem of dynamic scene in real world. Finally, we propose a generative model to deal with the problem of saturated reference — clipped pixels in the reference are either too dark or bright to allow for a reliable alignment.

The remaining chapter is organized as follows: we will give a brief review of the State of the Art in Section 2.2. In Section 2.3 to 2.5, we describe the three different approaches and show the benefits of our approaches by means of comparison with existing approaches.

2.2 Previous Work

The term, "HDR" we use in the thesis, refers to the process of merging several differently exposed pictures to produce one picture that includes all details from highlights to shadows. This is also known as modern HDR which first appeared with the advent of the digital camera. Before that, HDR focused on how to selectively increase or decrease the exposure of regions of the photograph to yield better tonality reproduction, such as *dodging and burning*, because one single shot of film camera can capture up to 4 orders of magnitude of brightness, close to that of human vision in a single view.

The idea of combing LDR images dates back to the pioneering works by Mann and Picard (1995) and Debevec and Malik (1997). A common way is to compute a weighed average over intensities or irradiance from a given exposure stack, and the weights are determined by the noise characteristics of the sensor (Akyüz and

Reinhard, 2007; Hasinoff et al., 2010; Granados et al., 2010). When the same pixel across the stack captures irradiance from different objects in the scene, whether because of camera motion or a moving object in the scene, it generates ghosting artifacts because the averaging process produces transparent copies of the moving objects.

Much research has been proposed to address the ghost artifacts. In general, these approaches can be summarized into two categories as follows:

2.2.1 Ghost Detection

All these approaches assume that the camera is static, or that a global registration of the background can be performed. Instead of aligning the scene, they detect the ghosted regions and only merge the information from the non-ghosted regions to generate final result. Gallo et al. (2009) model the exposure change and determine patches that might contain moving objects by counting the pixels that deviate from the predicted behavior. Raman and Chaudhuri (2011) follow a similar idea, but they model the intensity change and detect the motion in irregular patches obtained by grouping pixels into super-pixels. These algorithms pay for the reduction of motion artifacts with a potentially reduced dynamic range, as they drop data that does not follow the registration of the background.

Some algorithms incorporate the deghosting process in the weighting used to merge the pixels. Jacobs et al. (2008) detect pixels that would cause ghosting based on the variance and entropy across the exposure stack. Khan et al. (2006) use kernel density estimators to compute the probability that a pixel belongs to the background and weight the pixel based on the computed probability. Heo et al. (2011) use a weight that emphasizes well-exposed pixels and a second weight that enforces consistency across spatial and exposure domains. Zhang and Cham (2012) propose to weight the pixel using local gradients across the exposure stack as a measure of

consistency. While computationally efficient, these approaches have the drawback that they downweight or completely ignore pixels of moving objects except, possibly, in one of the images. At the same time, they may mix in inconsistent pixels, even if with a reduced weight.

2.2.2 LDR Alignment

A more elegant approach to address the artifacts due to the camera motion is to first align the LDR images. This task is complicated by the dramatic changes in brightness across the stack, since most registration algorithms rely on the *brightness constancy* assumption (Zitová and Flusser, 2003). Ward (2003) and Jacobs et al. (2008) address the brightness changes by binarizing each exposure and determining the optimal translation and rotation, respectively. Tzimiropoulos et al. (2010) compute the gradient map for each exposure and find a similarity transformation in the Fourier domain. Tomaszewska and Mantiuk (2007) use SIFT features to estimate a global homography. However, even assuming a perfectly static scene, such rigid transformations would be accurate only for planar scenes, or scenes where all the objects are far from the sensor's plane.

More sophisticated methods attempt to establish dense correspondences between the reference image and the other images in the stack. However, standard optical flow algorithms (Baker and Matthews, 2004) rely on the brightness constancy assumption, which is always violated, by construction, in the case of exposure stacks. Kang et al. (2003) boost the image intensity to compensate for this and use a standard optical flow to refine the correspondence mapping initialized by a global registration. Zimmer and Weickert (2011) propose to compute the optical flow in the gradient domain which is assumed to remain constant as exposures vary. The recent method by Sen et al. (2012) converts each image into a linear space inverting the camera response function, and selects an image as the reference for the final HDR image. Using a

variant of PatchMatch, they reconstruct an HDR image which maximizes the similarity with the reference image at the pixel level while minimizing the bidirectional similarity metric with the remaining images.

2.3 Admitting Brightness Inconstancy

Since the seminal works by Horn and Schunck (1981) and Lucas and Kanade (1981), optical flow has been widely used in a variety of applications in computer vision. Matsushita et al. (2005) apply optical flow to correct the motion caused by hand-shaking and generate stabilized video sequences. Liu and Freeman (2010) use optical flow to track pixels and produce a high-quality video by de-noising in temporal. The early HDR video work by Kang et al. (2003) and the most recent HDR alignment by Zimmer and Weickert (2011) also adopt the customized optical flow framework.

We start with optical flow for three reasons: First, it considers non-rigid and independent motion, while most other alignment techniques only can deal with a globally rigid transformation. Second, optical flow is a fundamental research problem in computer vision. Therefore, lots of existing observations and theoretical conclusions, such as optimization scheme, can be re-used in a direct or indirect way. Last but not least, its previous successful application in HDR alignment encourages the possibility of a generative model, although the previous works are limited to some specific situations.

In Section 2.3.1, we describe the conventional optical flow and its extended model to deal with *brightness inconstancy*. Later in Section 2.3.2, we describe the optimization algorithm to solve the proposed model. Experiments are shown in Section 2.3.3.

2.3.1 Model

Let R and S be two 2D images: $(\Omega \subset \mathbb{R}^2) \rightarrow \mathbb{R}^d$. For a gray-scaled image we have $d = 1$ and for color images $d = 3$. Moreover, $R(x, y)$ and $S(x, y)$ are then the

intensity value of the two images at the location $\mathbf{x} = [x, y]^T$, where x and y are the two pixel coordinates of a generic image point $\mathbf{x}$ in the image domain Ω. The image R will be referenced as the *reference* or *target* image, and the S as the *source* image. Let $\mathbf{u}(\mathbf{x}) = [u_x(\mathbf{x}), u_y(\mathbf{x})]^T$ be the flow for pixel $\mathbf{x}$ from S to R. The problem of conventional optical flow is formulated as:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \; E_D(\mathbf{u}) \; + \; \lambda \, E_S(\mathbf{u}) \tag{2.1}$$

where $E_D(\mathbf{u})$ is a score function that penalizes the differences of matched pixels between R and S. The prior term $E_S(\mathbf{u})$ is used to constrain the flow field and favor certain flow fields over others. $\lambda > 0$ is a regularization parameter that balances the contribution of data scores and prior.

$$E_D(\mathbf{u}) = \sum_{\mathbf{x}\in\Omega} \psi\big(R(\mathbf{x}) - S(\mathbf{x} + \mathbf{u}(\mathbf{x}))\big) \tag{2.2}$$

$$E_S(\mathbf{u}) = \sum_{\mathbf{x}\in\Omega} \psi\big(\nabla\mathbf{u}(\mathbf{x})\big) \tag{2.3}$$

where $\psi(\cdot)$ will be a penalty function to account for measurement error. There are many choices of $\psi(\cdot)$. See Baker et al. (2010) for details. In this chapter, we choose the differential version L_1 norm $\psi(s) = \sqrt{s^2 + .00001^2} \approx ||s||_1$ which allows us to deal with outliers or other non-Gaussian deviations of the matching criterion (Brox et al., 2004).

The above data scores defined in Eq. 2.2 are derived from the *brightness constancy* assumption $R\big(\mathbf{x}\big) = S\big(\mathbf{x} + \mathbf{u}(\mathbf{x})\big)$ as the pixel $\mathbf{x} \in \Omega$ flows from R to S. In many situations, this assumption is violated and in order to compensate the brightness changes, we introduce a more generalized data function as follows:

$$E_D(\mathbf{u}) = \sum_{\mathbf{x}\in\Omega} \psi\Big(\tau\big(R(\mathbf{x})\big) - S\big(\mathbf{x} + \mathbf{u}(\mathbf{x})\big)\Big) \tag{2.4}$$

where $\tau(r)$ is a function which accounts for how to transfer brightness in one image into the other image. Grossberg and Nayar (2003) name this as *brightness transfer function* (BTF). Note that Eq. 2.4 involves the bias and gain model as $\tau(r) = (1 + \alpha)r + \beta$ (where α is the gain and β is the bias). Under some situations, the brightness change can not be described by one function. For instance, strong lighting changes or flash/non-flash image stacks. In this dissertation, we are only interested in the scenes where brightness transfer could be described by one common global function, in particular, the brightness transfer in multiple exposures of the same scene.

Let us investigate more on the brightness transfer function before proposing our algorithm for solving Eq. 2.1. Suppose the exposure ratio between the reference image R and source image S is k and let f be the camera response function that relates the measured intensity to the scene irradiance, then the brightness transfer function from R to S can be described as follows:

$$\tau(r) = f\big(kf^{-1}(r)\big) \tag{2.5}$$

See Appendix A for a detailed derivation of Eq. 2.5. We neglect the details here but point out one important property for the brightness transfer function: $\tau(r)$ is monotonic increasing, which is important in designing the numerical algorithm in Sec. 2.3.2. Note that the hard monotonicity constraint is not a heuristic prior knowledge but a physical property determined by camera imaging pipeline (Grossberg and Nayar, 2003). Therefore, our final problem is formulated as:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \; E_D(\mathbf{u}) \; + \; \lambda \, E_S(\mathbf{u}) \tag{2.6}$$

$$\text{subject to: } \tau'(r) \geqslant 0$$

where $E_D(\mathbf{u})$ and $E_S(\mathbf{u})$ are defined in Eq. 2.4 and Eq. 2.3 respectively.

2.3.2 Algorithm

We propose to decompose the optimization problem in Eq. 2.6 into two relatively simple sub-optimizations, and then iterate between them until convergence. In the beginning stage, we initialize $\tau(r)$ using the intensity histograms of the images (Grossberg and Nayar, 2003) without geometric correction and initialize $\mathbf{u}$ as a zero flow field.

In the first step, given the existing $\tau(r)$, we optimize for $\mathbf{u}$. The sub-optimization becomes the conventional optical flow and we optimize it using the existing optical flow algorithm by Sun et al. (2010).

In the second step, given the existing $\mathbf{u}$, we seek to find the optimal solution for

$$\tau^* = \arg\min_{\tau} \sum_{\mathbf{x}} \psi\Big(\tau\big(R(\mathbf{x})\big) - S_{\mathbf{u}}(\mathbf{x})\Big) \tag{2.7}$$

$$\text{subject to: } \tau(r)' \geqslant 0$$

where $S_{\mathbf{u}}(\mathbf{x}) = S\big(\mathbf{x} + \mathbf{u}(\mathbf{x})\big)$ is an intermediate variable. The estimation of the brightness transfer function in Eq. 2.7 has been well studied because of its application in computer vision and image processing. A most commonly used solution is to model τ as a parametric model (Mann, 2000; Tico and Pulli, 2009; HaCohen et al., 2011). Instead of polynomial or cubic spline for τ, we represent it as piecewise cubic Hermite splines for two main reasons. First, cubic Hermite spline is more stable w.r.t small perturbations of the sample; Second, and the most importantly, cubic Hermite spline preserves the monotonicity of the samples. Let $\mathbf{p}$ be the vectorized parameters for τ, then Eq. 2.7 can be reformulated as:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \psi(\mathbf{A}\mathbf{p} - \mathbf{b}) \tag{2.8}$$

$$\text{subject to : } \mathbf{T}\mathbf{p} \geqslant 0$$

where $\mathbf{A}, \mathbf{b}$ and $\mathbf{T}$ are matrices encoding the data in Eq. 2.7. See Appendix B for more details about the derivation from Eq. 2.7 to Eq. 2.8. Note that the problem

is a quadratic programming problem if $\psi(s)$ is L_2 norm, but we defined $\psi(s)$ to be the differential version L_1 norm $\sqrt{s^2 + .0001^2}$ to limit the effect of outlier. However, computationally, we can introduce a sequence of quadratic models to approach the optimal solution for the problem in Eq. 2.8.

$$\mathbf{p}_k = \arg\min_{\mathbf{p}} (\mathbf{A}\mathbf{p} - \mathbf{b})^T \mathbf{W}_k (\mathbf{A}\mathbf{p} - \mathbf{b}) \tag{2.9}$$

$$\text{subject to : } \mathbf{T}\mathbf{p} \geqslant 0$$

where $\mathbf{W}_k$ is a diagonal matrix with $\mathbf{W}_k(i,i) = \psi\big(\Delta(i)\big)^{-1}$, $\Delta = \mathbf{A}\mathbf{p}_{k-1} - \mathbf{b}$ and the initialized value $\mathbf{W}_0 = \mathbf{I}$. As k approaches to infinite, the *iteratively reweighed least square*(IRLS) converges to the problem in Eq. 2.8, so does the sub-optimal sequence $\{\mathbf{p}_k\}$ converge to the solution for the original problem. The problem in Eq. 2.9 is a standard weighted quadratic programming with linear constraint and many scientific packages (such as MATLAB) can be used to solve it efficiently.

Algorithm 1 Optical Flow for Brightness Inconstancy

```
Initialization
for scale = coarse to fine do
    repeat
        Find a flow field u that minimizes Eq. 2.6 given existing τ
        repeat
            Find a BTF τ that minimizes Eq. 2.9 given existing u and τ
        until The brightness transfer function τ converges
    until The flow field u converges
    Upsample u to the finer scale
end for
```

Algorithm 1 gives an overview of our algorithm. Since $E_D(\mathbf{u}) + \lambda\, E_S(\mathbf{u})$ definitely decreases after each alternating step, it is locally convex and its lower boundary is

above zero, the proposed algorithm must converge to a static point. To avoid the unfavorable local minima, our algorithm is embedded into the coarse-to-fine scheme.

2.3.3 Results

For a quantitative comparison, we ran the method on all 8 sequences of the Middlebury benchmark with public ground truth(Baker et al., 2010). We used a score function as below to evaluate our method:

$$score(\mathbf{u}) = \frac{\sqrt{\sum_{\mathbf{x}\in\Omega} ||\mathbf{u}(\mathbf{x}) - \mathbf{u}^{GT}(\mathbf{x})||_2}}{|\Omega|} \tag{2.10}$$

where $\mathbf{u}^{GT}$ represents the ground truth flow field. The smaller the score is, the better the $\mathbf{u}$ is. Since all examples in Middlebury benchmark are almost brightness constant sequences, the results could not show the advantage of our method for brightness inconstant sequences. Therefore, we also ran our method on 8 synthesized sequences based on Middlebury benchmark. For each image pair in Middlebury training dataset, we applied a gamma correction of factor 2.2 to the second image to simulate non-linear brightness change.

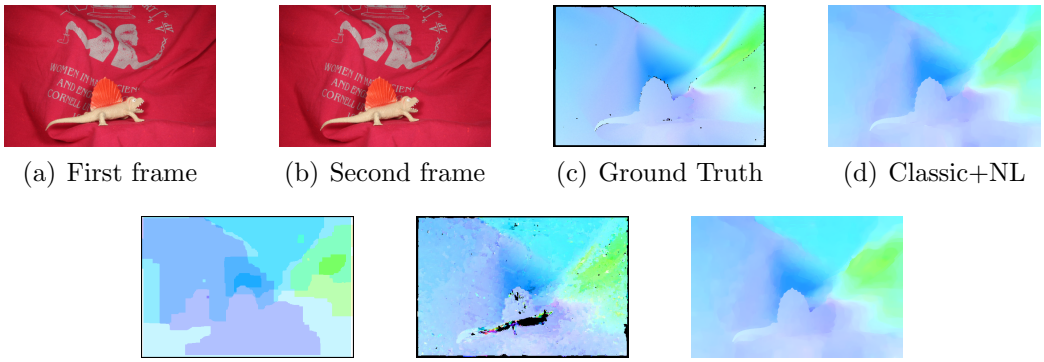
We compared our method with existing state-of-the-art dense correspondence methods designed for brightness inconstant scene: SIFT-Flow(Liu et al., 2008) and No-Rigid Dense Correspondence(NRDC) (HaCohen et al., 2011), as well as the **Classic + NL** by Sun et al. (2010) designed for brightness constancy case. For all these methods, we used the default parameter values suggested by the authors using their code. SIFT-Flow and **Classic + NL** generate a complete dense flow field, the average endpoint error(EPE) Eq. 2.10 is computed based on all pixels. For NRDC, a dense but incomplete flow field is estimated, so we compute EPE only based on the pixels whose flow are available.

Table 2.1: Average endpoint error(EPE) on Middlebury training set for Classic + NL, SIFT flow, NRDC and our model.

	Dimetodon	Grove2	Grove3	Hydrangea	RubberWhale	Urban2	Urban3	Venus
Classic + NL	0.107	0.1033	0.4704	0.1509	0.0736	0.2189	0.3755	0.2370
SIFT Flow	0.4357	0.5443	1.0812	0.3963	0.3674	1.3539	1.3929	0.4886
NRDC	0.2487	0.3577	0.8624	0.3245	0.4070	0.7065	1.9199	0.928
Ours	**0.186**	**0.0958**	0.5640	0.1758	0.192	**0.2166**	0.4658	0.4529

Table 2.2: Average endpoint error(EPE) on synthesized Middlebury training set for Classic + NL, SIFT flow, NRDC and our model.

	Dimetodon	Grove2	Grove3	Hydrangea	RubberWhale	Urban2	Urban3	Venus
Classic + NL	59.9574	50.2362	45.6593	77.5742	23.2609	42.1009	50.1077	17.666
SIFT Flow	0.4635	0.5578	1.180	0.4062	0.4010	1.4453	1.2909	0.6049
NRDC	0.223	0.4359	0.9231	0.3351	0.5300	0.8640	2.0886	1.0421
Ours	**0.101**	**0.184**	**0.6495**	**0.1789**	**0.138**	**0.214**	**0.8513**	**0.4811**



(a) First frame (b) Second frame (c) Ground Truth (d) Classic+NL

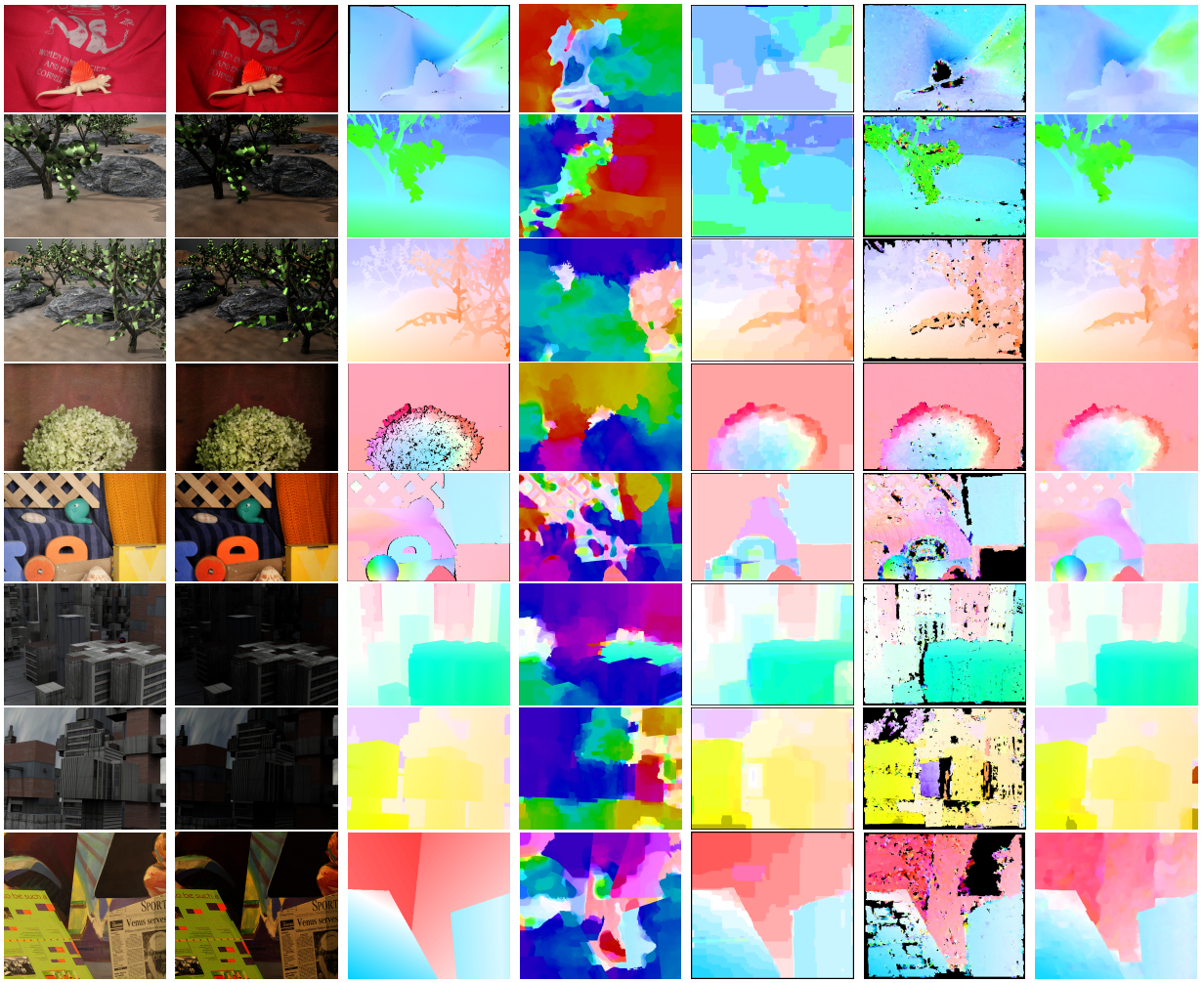
(e) SIFT Flow (f) NRDC (g) Ours

FIGURE 2.1: Results on Dimetrodon dataset from the Middelbury benchmark. Our method achieves results as competitive as **Classic + NL**, and is much better than SIFT-Flow and NRDC.

For brightness constant sequences, the hidden brightness transfer function is the identity function. In theory, if τ is correctly estimated by our algorithm, the performance of our method would be approximately the same that of **Classic+NL**, because the flow estimation embedded in our method is **Classic+NL**. Considering the numerical errors in optimizing τ, the performance of **Classic+NL** should be the upper bound performance of our method. Table 2.1 shows a quantitative comparison between our method, **Classic + NL**, SIFT-Flow and NRDC. While SIFT-Flow and NRDC perform worse than **Classic + NL** and our method in all examples, is as competitive as **Classic + NL** in most cases. Although the performance degradation caused by the numerical error in theory, our method performs better than **Classic + NL** in three of eight examples. A possible explanation is that the hidden brightness transfer function is only an approximate identity function, but our estimated τ provides a more accurate modeling. Figure 2.1 provides a visual comparison of flow generated by different methods. SIFT Flow suffers apparent discretization artifacts caused by the discrete optimization strategy. NRDC fails to compute a full dense flow and suffers artifacts caused by the randomized optimization startegy. Our method typically performs as good as **Classic + NL** in both visual and quantitative

comparisons.

Table 2.2 shows a quantitative comparison between different methods of the synthesized brightness inconstant sequences. Comparing with table 2.1, the performances are degraded for all these methods. The performance degradation of SIFT Flow, NRDC and our model is limited, but **Classic + NL** performs poorly because of the violation of the brightness constant assumption. Our method outperforms all other methods in all examples. Figure 2.2 provides a visual comparison. SIFT Flow and NRDC still suffer different artifacts caused by discretization and randomized optimization, respectively. **Classic + NL** almost failed for the failure of assumption, while our method typically performs robust to brightness changes in both visual and quantitative comparisons.



(a) First frame (b) Second frame (c) Ground Truth (d) Classic + NL (e) SIFT-Flow (e) NRDC (f) Our Method

FIGURE 2.2

2.4 Lifting the Displacement Constraint

In the previous Section, we extend the conventional optical flow to deal with *brightness inconstancy*, in particular, for the brightness transfer of exposure stacks in HDR. The primitive experiments show that it is a worthwhile solution. However, as conventional optical flow, one of the major practical limitations is that it only applies to the case of small motion. The most common optimization technique for optical flow is to linearize the data term (Eqn. 2.2) w.r.s the velocity, then apply the gradient descent algorithm or variational approach to solve the simplified linear problem. Nonetheless, in the case of large scale motion, the linearization based on the first order Taylor expansion becomes invalid.

In recent works, several authors propose several new techniques to deal with the large scale motion. Brox and Malik (2011) address the problem by combining the optical flow model with feature tracker which is reliable w.r.t large motion, they introduce an additional regularization term to enforce the consistency between dense optical flow and feature tracker on the tracked sparse features. Another approach proposed by Steinbr et al. (2009) is to decouple the data and prior term throughout a set of auxiliary velocity fields and approach the solution for original problem by a sequence of alternating sub-optimization. Our new model adopts the decoupling idea, but different from Stenibr *et al.* no auxiliary fields are introduced so less nested optimization exists. Moreover, we replace the pixel based data term in Eqn. 2.4 with a patch based data term. The same technique has been used by Lucas and Kanade (1981) to deal with *aperture problem*. We use it for the same reasons, and more or less, such patch (or block) based algorithms (SIFT flow and NRDC) have shown competitive performances in our previous experiment.

2.4.1 Model

We keep the same notation conventions as the previous section and introduce a new notation $\mathbf{P_x}$ that denotes a $p \times p$ patch centered at pixel $\mathbf{x}$. Let us start by replacing the pixel based data term with the patch based data term as follows:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} E_D(\mathbf{u}) + \lambda E_S(\mathbf{u})$$

$$\text{subject to: } \tau'(r) \geqslant 0$$

with

$$E_D(\mathbf{u}) = \sum_{\mathbf{x}\in\Omega} \psi\Big(\tau\big(R(\mathbf{P_x})\big) - S\big(\mathbf{P_{x+u(x)}}\big)\Big) \tag{2.11}$$

$$E_S(\mathbf{u}) = \sum_{\mathbf{x}\in\Omega} \psi\big(\nabla\mathbf{u}(\mathbf{x})\big) \tag{2.12}$$

Note that the minimizer for $E_D(\mathbf{u})$ is independent at each pixel. For each pixel $\mathbf{x}$, its optimal solution is the vector $\mathbf{u}$ that points the patch $\tau\big(R(\mathbf{P_x})\big)$ to the most 'similar' patch $S\big(\mathbf{P_{x+u(x)}}\big)$ from S. Likewise, $E_S(\mathbf{u})$ is a quadratic function of $\mathbf{u}$ and its optimal solution is any constant motion field. That is why the minimizer of $E_D(\mathbf{u}) + E_S(\mathbf{u})$ strongly encourages *piecewise smoothness* as λ is large enough. However, the trick in tuning the best λ is alway annoying for many practical problems. Moreover, such regularization framework always becomes substantially more cumbersome because of the explicit communication between the data term and prior term. Therefore, we need to find an alternative way to introduce this prior property without an implicit communication with the data term.

Instead of using *piecewise smoothness* as a *preferred* property of the optimal solution, we can enforce it as a *required* property. In real world, object surface can be approximated by piecewise planar and therefore for each pixel $\mathbf{x}$, there must exist a neighboring pixel, $\mathbf{y}$, that shares 'consistent' motion. Conventionally, the

consistency can be measured as the distance between $\mathbf{u}(\mathbf{x})$ and $\mathbf{u}(\mathbf{y})$ as follows:

$$\min_{\mathbf{y}\in\mathbf{N}(\mathbf{x})} ||\mathbf{u}(\mathbf{y}) - \mathbf{u}(\mathbf{x})||_2 < w, \quad \forall \mathbf{x} \in \Omega \tag{2.13}$$

where $\mathbf{N}(\mathbf{x})$ is the set of four neighbors of pixel $\mathbf{x}$ and w is the maximal motion bias for neighboring pixels from the moving object. w depends on the several factors: camera internal and relative external parameters and the minimal distance between camera center to the plane that the pixel $\mathbf{x}$ and $\mathbf{y}$ are back-projected to in the real world. In most situations, w is less equal to 1.

Now our new model can be formulated as follows:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \; E_D(\mathbf{u}) \tag{2.14}$$

$$\text{subject to: } \tau'(r) \geqslant 0$$

$$\forall \mathbf{x} \in \Omega, \; \min_{\mathbf{y}\in\mathbf{N}(\mathbf{x})} ||\mathbf{u}(\mathbf{y}) - \mathbf{u}(\mathbf{x})||_2 < w$$

Although the new energy function only includes data term, the hard inequality constraint implicitly enforces *piecewise smoothness*. Mathematically, the problem becomes optimization in a convex solution space defined by the hard constraints. This is seemingly more complicated than before, but, as we will see in the next section, the existing algorithm can find a well accepted solution for this problem in a very efficient way, even in the presence of large displacement.

2.4.2 Algorithm

Again, we optimize Eqn. 2.14 by iteratively sub-optimizations between $\mathbf{u}$ and τ until convergence. Algorithm 2 gives an overview of the optimization procedure. In the initialization stage, we estimate $\tau(r)$ using the intensity histograms of the images (Grossberg and Nayar, 2003) without geometric correction. Later, a sequence of alternating sub-optimizations about $\tau(r)$ and $\mathbf{u}$ respectively. Given the $\mathbf{u}$, the sub-optimization about $\tau(r)$ is the same as in the previous section; we also use the IRLS

method proposed in the section 2.3.2 to find the optimal $\tau(r)$. Given $\tau(r)$, the new sub-optimization is seemingly complicated, but we can adopt an approximate algorithm to solve it. We will discuss this later. Since the energy $E_D(\mathbf{u})$ definitely decreases after each sub-optimization stage, the optimization is indeed convergent.

Algorithm 2 Large Scale Non-rigid Correspondence

1: **Initialization**
2: **for** scale = coarse **to** fine **do**
3: **repeat**
4: Given existing τ, find a sub-optimal $\mathbf{u}$ of Eq. 2.14 using Algorithm 3
5: Given existing $\mathbf{u}$, estimate the sub-optimal BTF τ using IRLS algorithm
6: **until** The flow field $\mathbf{u}$ converges
7: Upsample $\mathbf{u}$ to the finer scale
8: **end for**

Given the latest updated $\tau(r)$, the sub-optimization about $\mathbf{u}$ in Eqn. 2.6 becomes:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \sum_{\mathbf{x}\in\Omega} \psi\big(R^\tau(\mathbf{P}_\mathbf{x}) - S(\mathbf{P}_{\mathbf{x}+\mathbf{u}(\mathbf{x})})\big) \tag{2.15}$$

$$\text{subject to: } \min_{\mathbf{y}\in\mathbf{N}(\mathbf{x})} ||\mathbf{u}(\mathbf{y}) - \mathbf{u}(\mathbf{x})||_2 < w$$

where $R^\tau(\mathbf{x}) = \tau\big(R(\mathbf{x})\big)$ is an intermedia variable representing the intensity mapped reference image.

Algorithm 3 describes the approximate algorithm we adopt to solve Eqn. 2.15. This algorithm is first proposed by Barnes et al. (2009) to locate the dense patch match between images. The algorithm begins with a randomized $\mathbf{u}$ and performs an iterative process of improving in scan order(from left to right, top to bottom), and each undergoes propagation stage followed by random search stage. In the propagation stage, $\mathbf{u}$ is improved by propagating the current vector flow from top

Algorithm 3 PatchMatch

1: Generate a random flow $\mathbf{u}$
2: **repeat**
3: **for** each pixel $\mathbf{x}$ from left to right, top to bottom **do**
4: Propagation: Improve $\mathbf{u}(\mathbf{x})$ using left and top neighbors‘ flow
5: Randomization: Improve $\mathbf{u}(\mathbf{x})$ by local random sampling
6: **end for**
7: **for** each pixel $\mathbf{x}$ from bottom to top, right to left **do**
8: Propagation: Improve $\mathbf{u}(\mathbf{x})$ using bottom and right neighbors‘ flow
9: Randomization: Improve $\mathbf{u}(\mathbf{x})$ by local random sampling
10: **end for**
11: **until** The flow field $\mathbf{u}$ converges

and left neighbors to $\mathbf{x}$ itself:

$$\mathbf{u}(\mathbf{x}) = \arg\min_{\mathbf{u}\in\mathcal{N}} \psi\big(R^{\tau}(\mathbf{P}_{\mathbf{x}}) - S(\mathbf{P}_{\mathbf{x}+\mathbf{u}(\mathbf{x})})\big)$$

$$\mathcal{N} = \{\mathbf{u}(\mathbf{x}), \mathbf{u}(\mathbf{x} - [1,0]^T), \mathbf{u}(\mathbf{x} - [0,1]^T)\}$$

In the random search stage, $\mathbf{u}$ is improved by a sequence of randomized search in local window,

$$\mathbf{u}(\mathbf{x}) = \arg\min_{\mathbf{u}\in\mathcal{W}} \psi\big(R^{\tau}(\mathbf{P}_{\mathbf{x}}) - S(\mathbf{P}_{\mathbf{x}+\mathbf{u}(\mathbf{x})})\big)$$

$$\mathcal{W} = \{\mathbf{u}(\mathbf{x} + w(.5)^i \mathbf{r}_i)\},\ i = 1, \cdots,$$

where w is the maximal search radius and $\mathbf{r}_i$ is a uniform random vector in $[-1,1] \times [-1,1]$. The random search stage terminates as $||w(.5)^i \mathbf{r}_i||_2 < 1$. The same iterative process also applies in the inverse-scan order (from bottom to top, right to left).

A rigorous proof of convergence of this algorithm and some other relative properties are out the scope of our thesis. For more details, please refer to Barnes et al. (2009). The key insights behind the algorithm are that some good patch matches can be found via random sampling, and that natural coherence in the imagery allows to propagate such matches quickly to surrounding areas.

We adopt this algorithm with three main reasons: First, the random search allows large displacement. Different from the optical flow, no implicit or explicit constraint is enforced to limit the magnitude of the velocity. Second, the propagation stage implicitly encourages the piecewise smoothness. Although the random search stage might violate the local smoothness constraint it is likely to be enforced again in the next iteration because of the natural coherence in the image. Last but not least, the algorithm is conceptually simple and very efficient in practice. It has shown a variety of applications in interactive image editing (Barnes et al., 2010).

2.4.3 Results

In this section, we show the performance of the proposed algorithm on a number of challenging stacks for HDR. For each case, given the input exposure stack, our method produces a new aligned stack and we use the Exposure Fusion method by Mertens et al. (2007) to generate the final HDR result. We start by providing a comparison with state-of-the-art approaches for globally pre-registered images. While this is an arguably simpler problem, and there exist algorithms showing good results in such a case, it is an important benchmark and allows to illustrate some strengths of the proposed approach.

Gallo et al. (2009) choose a reference image from the stack, divide it in patches, and combine patches from the other images only if they are consistent with the reference itself, without attempting to perform any registration. While their results are visually pleasing, we claim that such a strategy discards valuable information.

Figure 2.3 shows a comparison with their method. It also compares how many of the original pictures have been used for each pixel of the final result. Note that for large portions of the image Gallo et al. use only one or two images, discarding blocks both due to motion and over/under saturation, while the proposed algorithm combines more pictures from the stack, only discarding too bright or dark pixels. This provides a higher contrast in some areas, such as the dead tree trunk on the left of the image.

Zhang and Cham (2010) address the ghosting problem by weighting corresponding pixels in the stack based on the alignment of their gradients. They show that their method works well on a number of examples; but some artifacts are not completely removed, as is evident in the clouds in Figure 2.4. Rather than non-rigidly registering the images in the stack, they attenuate or discard pixels that would produce artifacts. This is the reason why the trees look washed-out. It should be said that, for stacks larger than 3-4 images, their method allows to remove objects appearing in one location in only one image, sometimes a desirable feature.

Kang et al. (2003) and Zimmer and Weickert (2011) are more similar to our approach in that they attempt to recover the non-rigid pixel transformations between shots. They both propose elegant solutions to the problem; however, they largely rely on the quality of the optical flow; whereas, we carefully evaluate when the optical flow works and have a recovery strategy for when it fails. Figure 2.5 shows a comparison with Kang et al. While their results are visually pleasant, some artifacts remain that are caused by mistakes of the optical flow, as shown in the blow-ups in Figure 2.5(b). Similar considerations apply to the comparison with Zimmer et al. Figure 2.6: When small objects or people are moving, the optical flow estimation may fail, in which case the final image is affected by ghosting.

A particularly difficult case for HDR imaging is that of non-rigid objects changing their appearance. For example, a typical situation with which amateur photographers often struggle is that of scenes containing water. Waves and ripples on the surface of

the water are extremely difficult to register to begin with, and the different exposures exacerbate the problem. In Figure 2.7 we show one such scene and the result of using our algorithm. We also show the artifacts that the motion of the water causes by showing the result of fusing the three LDR images after aligning them with a rigid, global registration.

FIGURE 2.3: A comparison with Gallo et al. (2009), (left) and ours (right) for the inset marked with the red box. Note how our method produces a better contrast in the dark area of the trunk, thanks to the fact that we use a larger set of images for each pixel; Gallo et al. discard a whole patch if enough of the pixels it contains are detected as outliers. See text.

FIGURE 2.4: A comparison between the method by Zhang and Cham (on the left) and our algorithm. Note the artifacts due to ghosting in the clouds. Image stack courtesy of Wei Zhang.

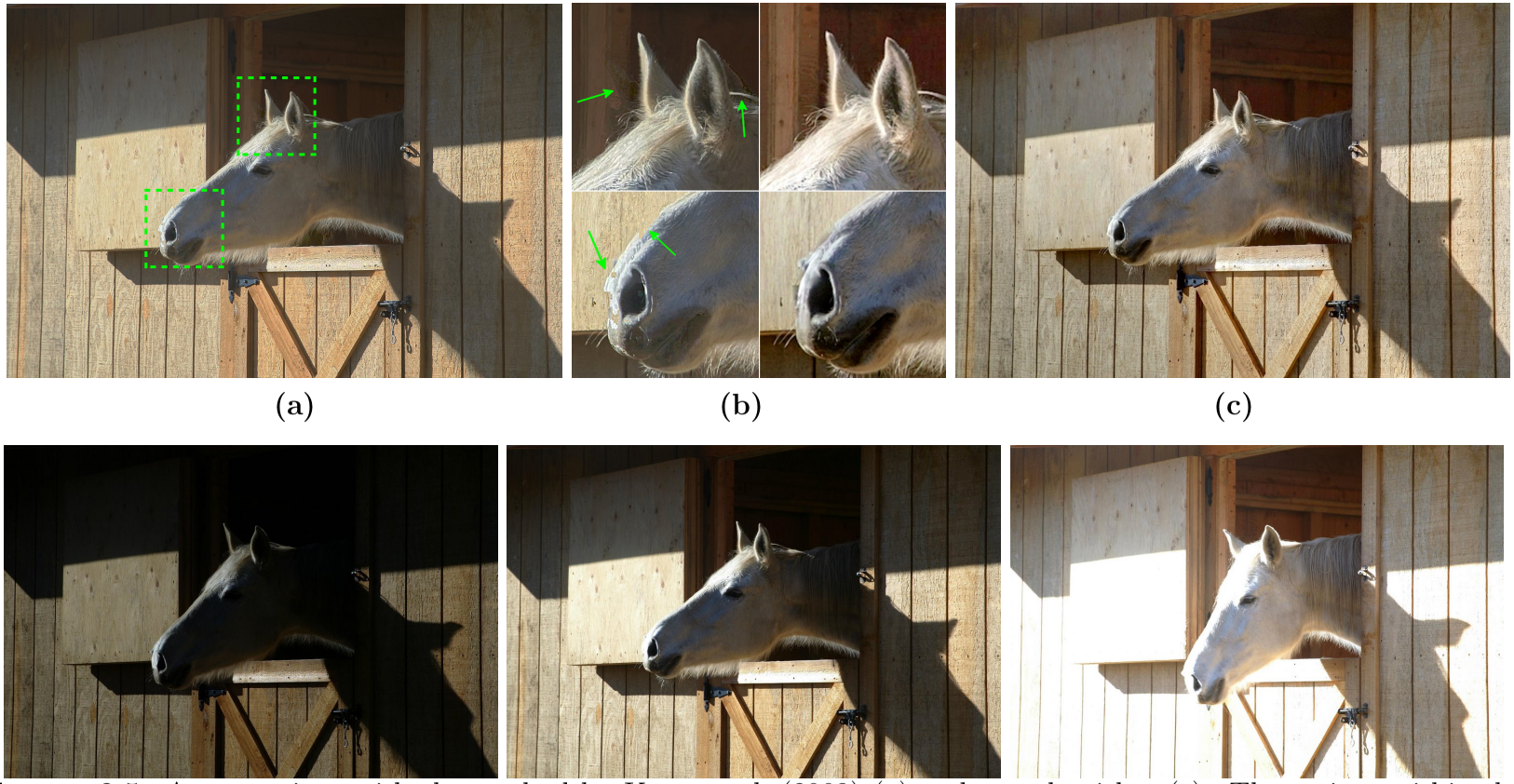


FIGURE 2.5: A comparison with the method by Kang et al. (2003) (a) and our algorithm (c). The regions within the green boxes are blown-up for comparison in (b), where the arrows point to the artifacts produced by the method by Kang et al. (left column). The bottom row shows the original images from the stack. Image stack courtesy of Sing Bing Kang.

FIGURE 2.6: A comparison with the method by Zimmer et al. On the left the full result of our algorithm, with the red blocks indicating the locations of the insets on the right. (a) and (e) are their results while (b) and (d) are ours. Notice how our method solves the ghosting problems apparent in the results by Zimmer *et al.*). Image stack courtesy of Henning Zimmer.

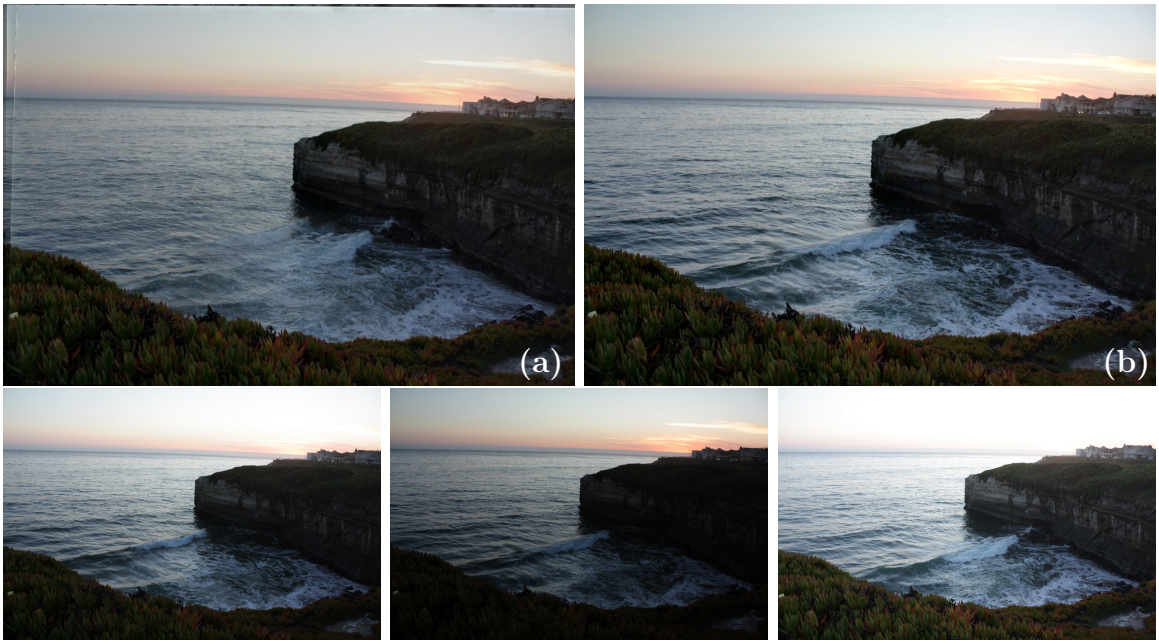


FIGURE 2.7: High-dynamic-range images of scenes containing water are notoriously difficult to capture due to the non-rigid motion of the water, as in the case of the LDR shots shown in the bottom row. A single, rigid homography only succeeds in registering the static parts of the scene (a), whereas our method (b) creates a picture that is as crisp as if the three LDR images used to generate it were taken at the same time.

2.5 Non-linear Stack Synthesis with Saturated Sensor Data

So far, the underlying assumption behind all previous works is *brightness constancy* under τ compensation, which is derived from *radiance constancy* (Appendix A).

$$S\big(\mathbf{x}+\mathbf{u}(\mathbf{x})\big) = \tau\big(R(\mathbf{x})\big) = f\big(kf^{-1}(R(\mathbf{x}))\big) \tag{2.16}$$

Mathematically, Eqn. 2.16 is equivalent to *radiance constancy* if the camera response function f is invertible. However, in many practical situations, because of several internal factors in the digital imaging pipeline, such as saturation charge and AD/compression quantization, f is invertible only in some ranges (See Figure 2.8). For the regions with exposure range with too low irradiance or too high irradiance, f is not invertible and results in loss details. Consequently, a reliable alignment for all these regions is extremely difficult.

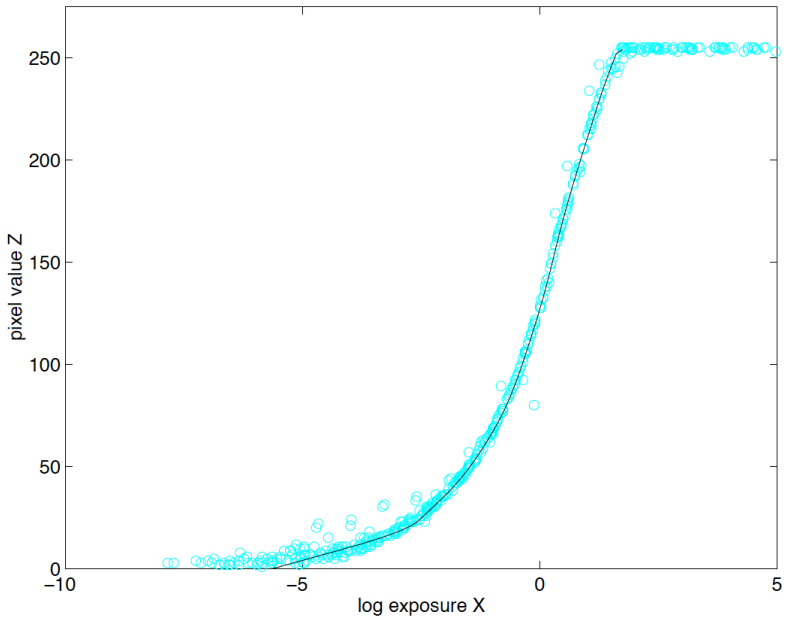


FIGURE 2.8: Typical camera response function, showing the mapping between incoming log irradiance (exposure) and output eight bit pixel value. Images courtesy of Debevec Paul.

The current state-of-the art method is the work by Sen et al. (2012). As many existing works (Kang et al., 2003; Zimmer and Weickert, 2011; Hu et al., 2012), Sen *et al.* first select an image R from the input exposure stack as the reference for the final HDR image, but instead of aligning the exposure stack, they consider HDR reconstruction as an image synthesis problem: For the regions where R is well exposed, they only use this information for final HDR reconstruction. For the remaining areas where R is clipped, they model this as a hole filling (image completion) problem and apply a patch-based algorithm to synthesize it, using information from the remaining image in the stack.

Our method follows a similar idea of synthesizing the content for the regions where R is clipped. However, unlike their work, our method can be applied to generic non-linearized exposure stacks, while Sen *et al.* only allow linearized exposure inputs. Moreover, our method attempts to use both intensity and gradient information from

all exposure stacks even for the region where R is well exposed, so we can preserve more details and also have less noise. Last but not least, our synthesis algorithm is guided by R with consistency check and, therefore, our method can produce more plausible results.

2.5.1 Model

Our method works by first selecting the image with the highest number of well-exposed pixels to be the *reference* image R (Kang et al., 2003; Gallo et al., 2009). Then, for each *source* image S in the stack, it synthesizes a new image L (the *latent* image) that looks like the reference image R, only exposed like S. In particular, the resulting latent image L has two important properties: First, where the reference R is properly exposed, L has image content that is geometrically compatible with R. In Figure 2.9, where the reference R is the middle exposure, this means, for instance, that the arms of the woman in the latent images L must appear in the same location as they appear in R. Second, if the reference R has areas that are either too dark or too bright to perform a reliable registration, the resulting L must have content from S that could plausibly appear there. For example, for areas that are saturated in the reference, such as some regions outside the window in Figure 2.9, we just need to find content from the source S that matches the neighboring areas (which we can reliably register) and whose luminance could plausibly match the bright pixels in the reference. The latent image L simply cannot be "too dark" there. If the reference had been the darkest image (top row in Figure 2.9), the areas posing these difficulties would have been the dark areas, where details are lost due to clipping.

Figure 2.10 illustrates our process in more detail. The reference R is on the left (red), the source S is on the right (blue), and we want to create a latent image L in the center (green), so the shapes of objects in L look like they do in R, except that they have the luminance range of S. We first initialize L by applying a color

mapping function τ to R, where τ is initialized using the intensity histograms of the images (Grossberg and Nayar, 2003), and is later refined as L is updated. We then find dense correspondences between L and S at the patch level using the Generalized PatchMatch algorithm (Barnes et al., 2010). If the reference patch $R(\mathbf{P_x})$ is not clipped; that is, it is mostly midtones and does not contain too dark or bright pixels, PatchMatch looks for a match from S. However, if $R(\mathbf{P_x})$ is clipped, neither the color mapping τ, nor direct registration is reliable. In this case, we modify the PatchMatch to find a patch $S(\mathbf{P_y})$ that could plausibly match $R(\mathbf{P_x})$: pixels in $S(\mathbf{P_y})$ should match the pixels in $R(\mathbf{P_x})$ that are not clipped, and the rest of the pixels in $S(\mathbf{P_y})$ would clip under the current τ. Note, however, that those pixels don't necessarily clip in S, allowing us to bring in more detail to L than is available in R. As we progress, the intensity mapping function τ is updated and refined based on the dense correspondence. To avoid a bad local minimum and to better synthesize clipped areas, these processes are executed iteratively using a coarse-to-fine schedule. We now proceed to explain the details of the whole system.

FIGURE 2.9: Our method takes, as an input, an exposure stack of a dynamic scene captured with a hand-held device (left column, note the dramatic changes in the scene). It selects a reference image and, for each of the other images in the stack, synthesizes an image that looks as if it was taken at the same time as the reference, only with different exposure settings (middle column). These images can then be fused into a single image showing more details (right). Our approach allows gathering data from the images in the stack even for regions that are severely under/over exposed in the reference, a main limitation of many state-of-the-art approaches.

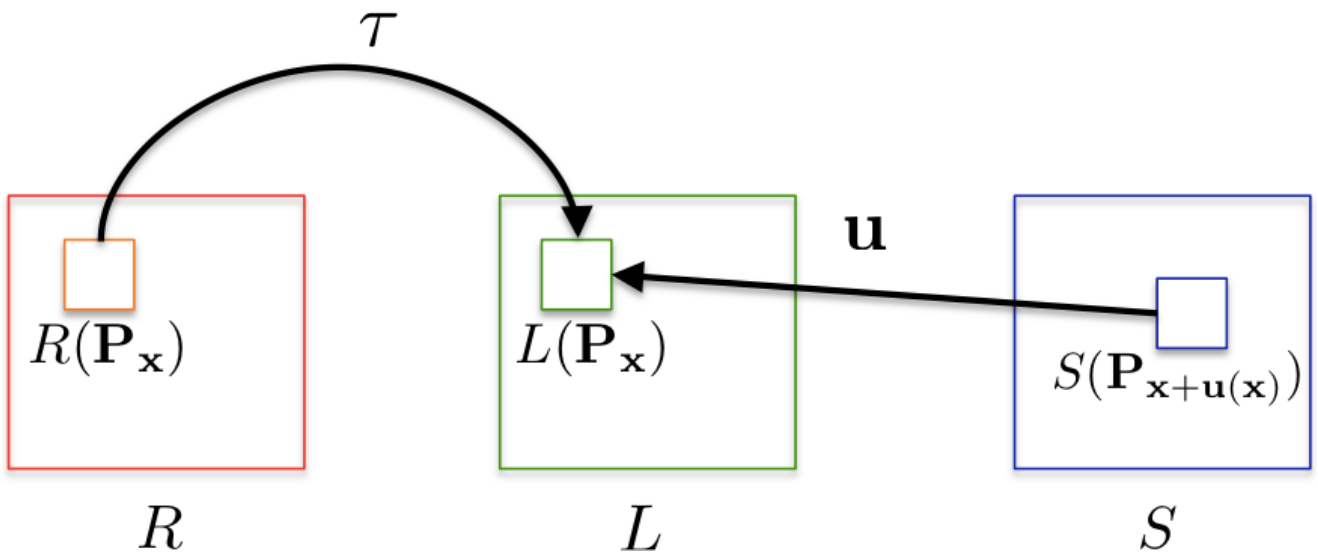


FIGURE 2.10: For each source image S in the stack, we want to synthesize the latent image L we would have, if we had captured it at the same time as the reference image R, but with the same exposure settings as used to capture S. τ is an intensity mapping function accounting for how the pixel values change under the exposure change. $T(\mathbf{P_x})$ is the $p \times p$ patch centered at pixel $\mathbf{x}$, and the $T \in \{R, L, S\}$ indicates the image it is from. $\mathbf{u}(\mathbf{x})$ maps pixel locations from L (and R) to S.

Two-picture Synthesis Model

We wish to synthesize the latent image L that looks just as if R was taken using the exposure setting of S: in other words, L should be consistent with R everywhere in geometry. To account for this, we need to define a *radiance consistency* measure C_r between two images. To maximize the applicability of our algorithm, we do not want to limit its scope to RAW (linear) images. Image signal processors (ISP) apply various non-linear transformations to the almost-linear pixel values; these highly non-linear transformations are usually much more sophisticated than simple gamma compressions, and they sometimes even depend on the image content (Hu et al., 2012; Kim et al., 2012), making it difficult, or even impossible, to invert the transformations. Hence, instead of linearizing the input images, we take inspiration from the energy definition by Darabi et al. (2012), but we account for a generic intensity mapping function τ:

$$C_r(L, R, \tau) = \sum_{\mathbf{x}\in\Omega} \left(\, d(L, \tau(R)) + \alpha \, d(\nabla L, \nabla\tau(R)) \, \right), \tag{2.17}$$

$$\text{subject to: } \tau'(r) \geqslant 0$$

where, for clarity, we omitted the dependency of R and L from the pixel location $\mathbf{x}$. Ω is the image domain and $d(x, y) = \|x - y\|^2$. For every pixel $\mathbf{x}$ in either image, we extract six channels: the three RGB components and the three corresponding gradients. The parameter α balances the color and gradient (texture) consistencies. In addition to boosting the details of the texture (Agarwala et al., 2004; Pérez et al., 2003), using gradients helps to compensate for exposure changes (Zimmer and Weickert, 2011). As in the previous sections, $\tau(r)$ is required to be monotonic.

Minimizing the cost of Eq. 2.17 is an ill-posed problem, so we need additional constraints to better define L. We can define a term structurally very similar to

Eq. 2.17, that encourages *texture consistency* between L and the source image S:

$$C_t(S, L, \mathbf{u}) = \tag{2.18}$$
$$\frac{1}{p^2} \sum_{\mathbf{x} \in \Omega} \Big(\, d\big(L(\mathbf{P}_\mathbf{x}), S(\mathbf{P}_{\mathbf{x}+\mathbf{u}(\mathbf{x})})\big) \; + \alpha \, d\big(\nabla L(\mathbf{P}_\mathbf{x}), \nabla S(\mathbf{P}_{\mathbf{x}+\mathbf{u}(\mathbf{x})})\big) \, \Big) \,,$$

where $S(\mathbf{P}_\mathbf{x})$ is a $p \times p$ patch centered at $\mathbf{x}$ in image S (same for $L(\mathbf{P}_\mathbf{x})$ and L) and $\mathbf{u}(\mathbf{x})$ maps patches in L to the corresponding patches in S, see Figure 2.10. If α is set to zero, this term resembles the coherence term defined by Wexler et al. (2007) and Kopf et al. (2012). We operate in the *RGB* color space and only search over translations, which makes the updates of L faster but does not lower the quality of our results, given the expected changes in an exposure stack.

Now we can address our new model as follows:

$$L^* = \arg\min_{L,\tau,\mathbf{u}} \{ \; C_r(L, R, \tau) + C_t(S, L, \mathbf{u}) \; \} \tag{2.19}$$
$$\text{subject to: } \tau'(r) \geqslant 0$$

2.5.2 Algorithm

We propose to decompose the optimization problem in Eq. 2.19 into three relatively simple sub-optimizations, and then iterate between them until convergence. Algorithm 4 gives an overview of the whole algorithm. For the coarsest level, we initialize τ using the intensity histograms of the images (Grossberg and Nayar, 2003) to minimize the effect of misalignment between the images, initialize $L = \tau(R)$, and apply the Generalized PatchMatch on S and L to initialize $\mathbf{u}$.

In the first step, given the existing L, we optimize for $\mathbf{u}$; note that $\mathbf{u}$ only appears in C_t in Eq. 2.19. C_t can be minimized globally with respect to $\mathbf{u}$, as the latter is independent for each pixel, see Eq. 2.18. The optimal solution can, therefore, be reduced to finding the nearest-neighbor patches in S for each patch $L(\mathbf{P}_\mathbf{x})$. Instead of a complete search, we use Generalized PatchMatch (Barnes et al., 2010).

Algorithm 4 Generative Model to deal with Saturation

1: **Initialization**
2: **for** scale = coarse **to** fine **do**
3: **repeat**
4: Given existing L, optimize C_t w.r.t **u** using Patch Math algorithm
5: Given existing **u** and τ, optimize L by solving screened Poisson equation
6: Given existing L, estimate the sub-optimal BTF τ using IRLS algorithm
7: **until** The flow field **u** converges
8: Upsample **u** to the finer scale
9: **end for**

In the second step, given the existing **u** and τ, we seek to find a solution L that minimizes Eq. 2.19. Note that this equation is quadratic in L and it is, therefore, equivalent to the following quadratic function:

$$L^* = \arg\min_L \{\ d(L,T) + \alpha\ d(\nabla L, \nabla T)\ \} \tag{2.20}$$

where, using Eqs. 2.17 and 2.18, and summing over the pixels in the patches rather than over the patches themselves, Eq. 2.18, we define the auxiliary image T as

$$T(\mathbf{x}) = \frac{1}{2}\tau\big(R(\mathbf{x})\big) + \frac{1}{2p^2}\sum_{\mathbf{y}\in N(\mathbf{x})} S\big(\mathbf{x}+\mathbf{u}(\mathbf{y})\big) \tag{2.21}$$

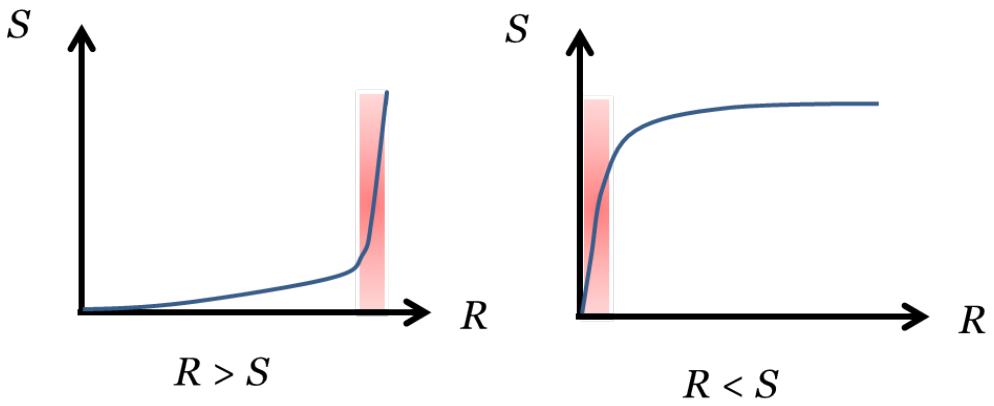
where $N(\mathbf{x})$ is a $p \times p$ window centered at $\mathbf{x}$. Basically, T is the weighted average of the colors of all the similar pixels in S and the patch in $\tau(R)$, while ∇T denotes the weighted average of the gradients. Eq. 2.20 is a Screened Poisson equation, which can be optimized efficiently in the Fourier domain (Bhat et al., 2008). The square function $d(x, y)$ we defined is computationally efficient, but the result is very sensitive

to outliers. To avoid the effects of outliers, we add two weighting terms:

$$T(\mathbf{x}) = \frac{1}{s}\left[w_\tau\big(\mathbf{x}\big)\tau(R(\mathbf{x})\big) + \frac{1}{p^2}\sum_{\mathbf{y}\in N(\mathbf{x})} w_{\mathbf{u}}(\mathbf{y}) S\big(\mathbf{x}+\mathbf{u}(\mathbf{y})\big)\right] \tag{2.22}$$

where $w_\tau(\mathbf{x})$ and $w_{\mathbf{u}}(\mathbf{x})$ reflect the confidence of the intensity mapping function τ and the geometric mapping $\mathbf{u}$ for pixel $\mathbf{x}$, respectively, and s is the normalization factor $w_\tau(\mathbf{x}) + \frac{1}{p^2}\sum_{\mathbf{y}\in N(\mathbf{x})} w_{\mathbf{u}}(\mathbf{y})$. Note that with two additional weighting terms, if a matching pixel is in a useful range in both R and S, we combine the information. If one of the two images is bad, we use the other image. Finally, if both images are bad, we use only the reference.

As we discuss in the beginning of this section, the brightness transfer function τ, which describes how the RGB values change from the reference to the source image, cannot be accurate across the whole range, due to saturation and under-exposure. For example, if S was captured with a shorter exposure time (darker) than R, and if the top of the range in the domain of R is saturated, τ will be flat in that area, thus not providing any relevant information; all the useful information for registration and HDR image creations in S.



The opposite may be true when S was captured with a longer exposure time, see inset, where red bands show the range in which the mapping τ is not reliable. We

choose w_τ to reflect the quality of $\tau\big(R(\mathbf{x})\big)$: $w_\tau(\mathbf{x})$ is ϵ (a small constant) if $R(\mathbf{x})$ is severely over/under exposed, but if the quality of pixel $\mathbf{x}$ is good, $w_\tau(\mathbf{x})$ is 1.

The weight function $w_{\mathbf{u}}(\mathbf{x})$ indicates the confidence in the mapping $\mathbf{u}(\mathbf{x})$. Ha-Cohen et al. (2011) define this confidence using the local consistency ratio of $\mathbf{u}(\mathbf{x})$, but this may fail for the regions where over- or under-exposure causes the texture to be weak. Instead, inspired by Wexler et al. (2007), we define:

$$w_{\mathbf{u}}(\mathbf{x}) = \begin{cases} \exp\left\{ -\dfrac{d\Big(\tau\big(R(\mathbf{P}_{\mathbf{x}})\big),\ S\big(\mathbf{P}_{\mathbf{x}+\mathbf{u}(\mathbf{x})}\big)\Big)}{2\sigma_1^2} \right\} & \text{if } R \text{ is not clipped} \\ \exp\left\{ -\dfrac{d\Big(R\big(\mathbf{P}_{\mathbf{x}}\big),\ \tau^{-1}\big(S(\mathbf{P}_{\mathbf{x}+\mathbf{u}(\mathbf{x})})\big)\Big)}{2\sigma_2^2} \right\} & \text{if } R \text{ is clipped} \end{cases} \tag{2.23}$$

Intuitively, we normally want to use pixels from S when they are consistent with R (the first case in Eq. 2.23). However, consider an area that is saturated in R and assume that we are working with an S that is darker, and, therefore, better exposed. In such regions, $\tau\big(R(\mathbf{P}_{\mathbf{x}})\big)$ is not reliable and we want to relax the requirement that patches from S have to match, or we would reject all the patches in that area. On the other hand, if a patch in S is so dark that it wouldn't possibly become saturated in R, we also don't want to allow its use. Basically, by applying τ^{-1} to $S(\mathbf{P}_{\mathbf{x}+\mathbf{u}(\mathbf{x})})$ first, (the second case in Eq. 2.23), we say that if this patch from S would saturate, we are still willing to use it. In this way, the clipped areas of R in L can be reasonably synthesized using the information from S. σ_1 and σ_2 are data-dependent parameters controlling the smoothness of the induced error surface and we compute them as Wexler et al. (2007).

In the third and last step, given the existing L, the optimization about BTF is the same as in section 2.3.2 using IRLS.

Multi-scale Solution

At each of the steps described above, the objective function is guaranteed to not increase. To further enforce a better local optimum, and to speed up convergence, we use a pyramid approach. The optimization starts at the coarsest scale of a Gaussian pyramid, and the solution is propagated to finer scales. When moving from one level to a finer one, three variables need to be propagated: we transfer $\tau(r)$ as is, and linearly interpolate the mapping $\mathbf{u}$. However, we found that linear interpolation of the latent image L leads to blurry results. Therefore, we propagate the solution using the weights w_τ and $w_{\mathbf{u}}$ described above. The rationale is that each pixel of image L at a given scale should be initialized with the corresponding pixel from the reference image from the same level of the pyramid (appropriately mapped with τ) if it is within a reasonable range. Otherwise, it should be initialized using the source image S (using the mapping u derived from the previous level).

2.5.3 Results

We now compare the performance of our algorithm to state-of-the-art approaches. As mentioned before, we are only aware of four methods that attempt to address the general case of camera motion and scene changes at the same time (Kang et al., 2003; Zimmer and Weickert, 2011; Hu et al., 2012; Sen et al., 2012).

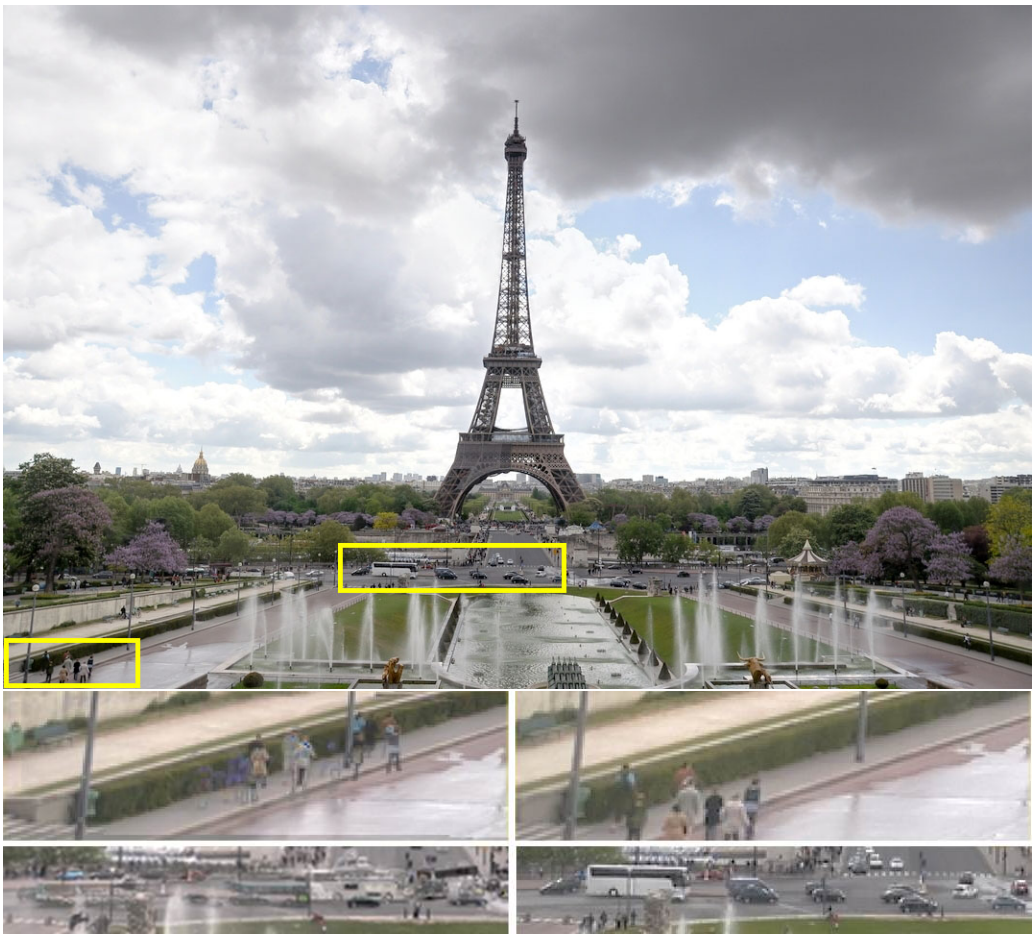


FIGURE 2.11: Comparison with Zimmer and Weickert (2011). The top image shows our result. The two bottom rows show blow-outs of two different regions of the image that are problematic for Zimmer *et al.* (left). Our algorithm does not produce artifacts (right). Images courtesy of Henning Zimmer.

FIGURE 2.12: Comparison with Kang et al. (2003). Note the large motion of the head in the original images (left). The middle image is the result by Kang *et al.* , note that the artifacts around the ears and muzzle. On the right is our result. Images courtesy of Sing Bing Kang.

All the fused results were generated using the method by Mertens et al. (2007), with the exception of Figure 2.14, which was tonemapped with the method by Mantiuk et al. (2006) to allow for a fair comparison with the method by Sen *et al.* Figures 2.11 and 2.12 show results sensibly better than Zimmer *et al.* and Kang *et al.* , respectively. When the reference image is reasonably well-exposed everywhere, our method produces very similar results as Hu *et al.* However, when part of the reference is saturated, as in Figure 2.13, Hu *et al.* , discard valuable information from the shorter exposure (first row, middle image). Our method, on the other hand, successfully captures all the available information in the synthesized latent image (second row, middle image). Sen *et al.* assumes exposure stacks of RAW or linearized images. For the examples shown in Figure 2.13, this assumption is violated because no estimation of the camera response function was available, and their result shows suffers many visible artifacts. Figure 2.14 shows another case with a large saturated region. We use RAW images as the input for the algorithm by Sen *et al.* and their non-linear counterpart (first row in Figure 2.14) as the input to our method. Note that the halos in the results by Sen *et al.* are not caused by the tonemapping algorithm, rather they are artifacts of their registration algorithm. In our result, (bottom, rightmost image in Figure 2.14) the sky is more faithful to the original images and no artifacts are introduced. As we mentioned in the previous section, we attempt to preserve as much information as possible from the exposure stack by using both the intensity and the gradients in our reconstruction.

As for any patch-based algorithm, our results are somewhat affected by the patch size p. In all our examples we used $p = 10$. However, in some situations this may be too small a neighborhood. Figure 2.15 shows an extreme case of a stack comprised of only two images, with a region that is saturated in both images, demonstrating one of the limitations of our method. Note that any existing method would be hard-pressed to achieve decent results here, because the dramatic change of exposure

makes it extremely difficult to match pixels across the images. The fact that the stack contains only two pictures also constitutes a challenging situation for most state-of-the-art algorithms. Our method can register the images correctly despite selecting a reference image that has a completely saturated sky. However, since the sun is saturated in both images, our algorithm fills in the saturated sun using non-saturated pixels from S. Since this region is saturated, the algorithm technically did the right thing: it filled the region with the available information. However, most photographers would prefer the sun to be left untouched; in this case a simple increase of the patch size to $p = 15$ solves the problem.

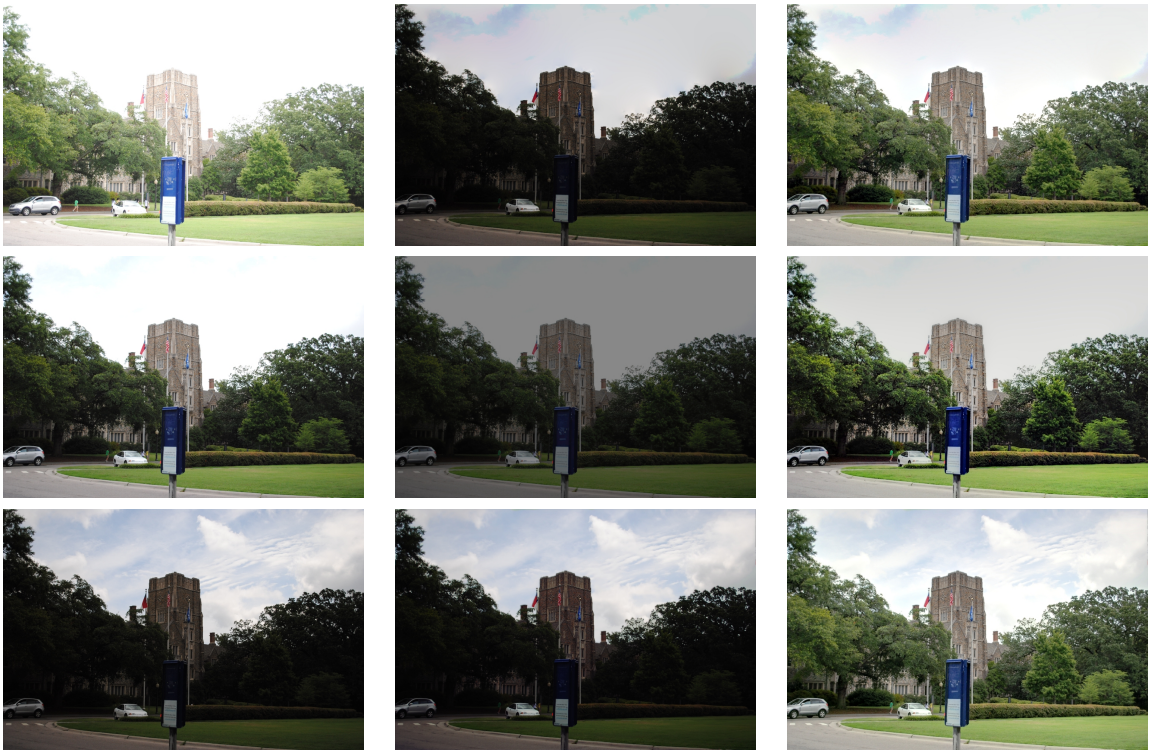


FIGURE 2.13: A comparison with Hu et al. (2012) and Sen et al. (2012). The first column shows the original images. The reference, as selected by Hu *et al.* , is the middle exposure. Notice that the sky is almost completely saturated, causing their algorithm to disregard useful information in the short exposure (top row, middle image), and leading to poor quality in the fusion result (top right). Sen's algorithm is designed to work on linear exposure stacks. For this non-linear stack, a reliable estimation of the camera response function would require acquiring a stack of registered images. Partially due to the non-linearity of the input images, their method fails in reconstructing the content for the saturated regions in the reference — both reconstructed HDR images (middle row, rightmost image) and the intermediate aligned shorter exposure stack (middle row, middle image) show a degraded quality. With the same reference frame our algorithm can synthesize a novel image which is completely consistent with the reference, and also captures all the details of the sky (bottom row, middle image). This directly reflects in the high quality of our exposure fusion result (bottom row, rightmost image).

FIGURE 2.14: Another comparison with Sen et al. (2012). The first column shows the original images in the stack; the middle exposure is selected as the reference. For the method by Sen *et al.* , we first linearize the original images and use the linearized exposure stacks as the input. Their algorithm generates a plausible result (top middle and right). However, their method still suffers from various artifacts. For example, the blurred sky in the saturated region and the halo around the dome are unexpected. Note that the halo in the reconstructed shorter exposure is not caused by tone mapping but the errors in HDR reconstruction. For the tone mapped HDR image (top right), the reconstructed sky is not natural for the saturated region in the reference. Our algorithm can synthesize an image (bottom middle) that is completely consistent with the reference and also preserves as much information as possible from the whole exposure stack. Our tonemapped HDR image is plausible and virtually artifact free (bottom right).

FIGURE 2.15: A very challenging 2-image stack. The original images (left) are dramatically separated in terms of exposure time — the areas that are correctly exposed in one are barely visible in the other. An interesting feature of this stack is that the region around the sun is saturated in both images. Note that the longer exposure, which we selected as the reference (left bottom), is completely saturated in the sky. Our algorithm attempts to synthesize the saturated region in the source image from other pixels in the same image, thus effectively removing the sun (middle top). A larger patch size ($p = 15$) forces the algorithm to leave the sun region untouched (middle bottom). The last column shows the exposure fusion result for the standard patch size (top) and for the larger patches (bottom).

3

Panoramic Stitching

You don't take a photograph, you make it.

Ansel Adams, 1902 - 1984

So far, we have been discussing how to produce a high dynamic range image using a series of images with different exposure. Another major limitation of digital camera is angle of view (also called field of view). As digital sensors attain progressively higher resolutions, and thereby smaller pixel sizes, the one quality of an image which does not benefit much is its filed of view (FOV). While each human eye individually has anywhere from 120-200 degree angle of view, a full frame 35mm standard camera with a standard lens (36-60mm) covers between 40 degree and 62 degree angle of view. Panoramic stitching enables us to create with a wider angle of view by capturing a series of image across all different angles and intelligently merging them together. This can be done with any digital camera and it will provide a much greater level of details and wider angle, which ordinarily only attainable with much more expensive equipment.

3.1 Introduction

Panoramic image stitching has been intensively studied and already has several commercially applications, standard stitching can be summarized as following stages:

Acquisition of images: Rotating camera in increments to encompass the desired field of view. The size of rotation increment the the number of totally sampled images depends on the angle of view for each image, which is determined by camera focal length and the amount of overlap between photos. Ideally, panoramic stitching algorithms require that camera rotates about the optical center of lens in order to avoid parallax error. Alternatively, the use of camera arrays (Wilburn et al., 2005; Horisaki et al., 2009; Brady et al., 2012) can snapshot images across all angles simultaneously and may enforce all cameras aligned to the same optical center in design, thereby avoiding parallax error intrinsically.

FIGURE 3.1: An instance of images captured through rotating camera to encompass a wider angle of view.

Alignment of images: Estimating camera matrix or homograph for each captured images such that all adjacent shoots are aligned. The literature methods for automatic image alignment fall into two categories: direct(pixel) and feature based. Direct methods are to shift or warp the images relative to each other and to look at how much the pixels agree. Since direct methods use all of available image data and hence they could provide accurate registration but they require a close initialization guess and also are computational expensive. The other major approach is to find distinctive features from each image, to match individual features to estimable a

global correspondence and to then estimate the geometric transformation between the images. Feature based methods are ease implementation and extremely fast, it has becomes standard algorithm in most image stitching softwares. The end results of this stage is a set of transformations which project each image to a common reference system in which all images are aligned and a virtual wide angle of view image is synthesized.

FIGURE 3.2: An instance of stitched image with seam blending (Bottom) and without seam blending (Top).

Blending of seams and deghosting: Eliminating the visibility of any seam between images due to exposure differences and reducing ghosting due to moving ob-

jects across adjacent images. The most commonly-used approach of *deghosting* is to discard the inconsistent pixel values, such as median filtering, or average them with content-aware weights (Shum and Szeliski, 1997; Uyttendaele et al., 2001; Peleg et al., 2001). An alternative way is to find optimal seam between regions where different images contribute to the final stitched panorama. Many heuristic energy-based functions are introduced to select the seam, the most sophisticated one is the graph-cut seams by Agarwala et al. (2004). Figure 3.3 shows that a seam blending typically provides a much pleasant result.

Cropping, touch-up and post-processing: Cropping the stitched panoramic image so that it adheres to a given rectangular image dimension. In some situations, post-processing such as tone mapping, color refinements and sharpening happen in this stage.

FIGURE 3.3: An instance of stitched image after cropping.

While digital image stitching offers a means of forming images over wide FOV without sacrificing image resolution, obtaining a high quality panorama is still difficult and relies on several favorable conditions: the captured scene being almost stationary (to avoid ghosting due to moving objects); overlap between adjacent im-

ages being roughly $10\% - 30\%$; and calibrated information on extrinsic and intrinsic parameters of the image acquisition system being fully or partially accessible. As we mentioned, one possible solution is to use camera arrays system to acquire data. With digital cameras becomes smaller and cheaper, constructing an array of camera becomes affordable and valuable because of its potential applications, such as wildlife habitat monitoring (Nichols et al., 2009), celestial exploration (Balme et al., 2012), and recognition or tracking of moving objects or people in crowded scenes (Gueguen et al., 2011).

In this thesis, we focus on image stitching algorithm for images captured with camera array system and in particular, we are interested in a novel micro-camera array - AWARE-2 - developed by Brady et al. (2012). Similar to many other camera array systems (Wilburn et al., 2005; Horisaki et al., 2009), AWARE-2 enables to fire all cameras simultaneously and allows each individual micro-camera to configure differently. Therefore, if all of the acquired images are in perfect alignment, the final stitched image will be mostly free of ghosting due to moving objects. Unlike existing camera array systems, AWARE-2 adopts a gigagon monocentric lens and arrange all micro-cameras along the focal surface of the objective (Figure 3.4), thereby lens speed and FOV can be scale independent and the final synthetic image could reach up to 50 gigapixels in theory (Brady et al., 2012).

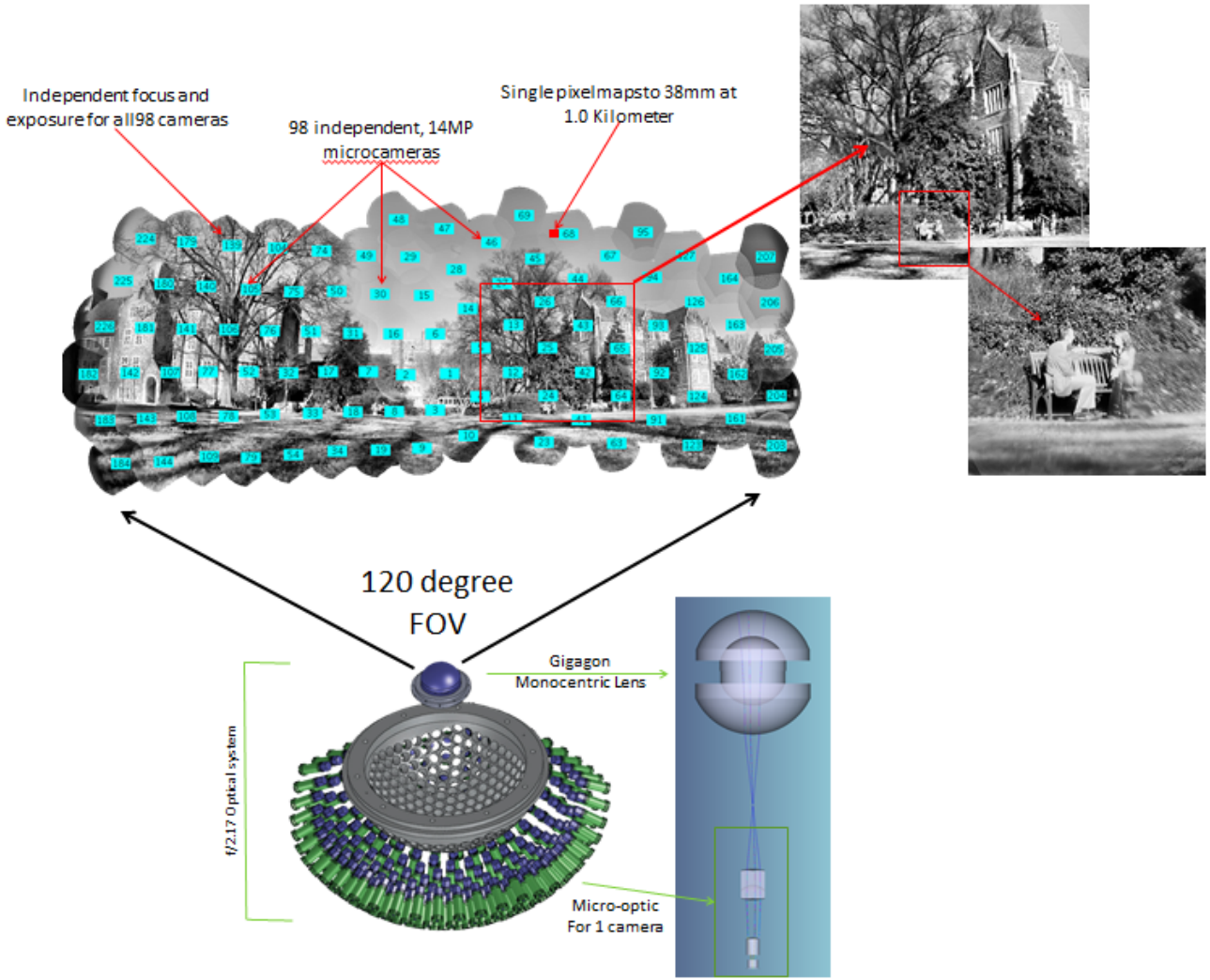


FIGURE 3.4: Outline of AWARE-2 Array Camera System, Image courtesy of Brady *et al.*

FIGURE 3.5: A mosaic of 7 AWARE-2 micro-camera images. Overlapping regions are highlighted in orange.

From an algorithm-design aspect, AWARE-2 has one main advantage: the constituent AWARE-2 snapshots can be treated as if shot in succession by a single camera capturing an *effectively* static scene, which is one major limitation for existing approaches. Nonetheless, several new problems arise with the 'optimal' optic design for maximizing the composite FOV and resolution. As is illustrated in Fig. 3.5, the adjacent micro-cameras have sparse, geometrically irregular and noisy (S.I.N) overlap. Most stitching algorithms require that the FOV configuration of camera arrays follows the conventional pattern of a regular grid (Wilburn et al., 2005; Kopf et al., 2007). Several algorithms can deal with grid-free image stacks but require significant and distinctive overlap between adjacent shots (Brown and Lowe, 2003; Brown et al., 2005). Therefore, the state of the art is considerably challenged by S.I.N condition in AWARE-2. Furthermore, the output raw images from AWARE-2 suffer from highly noisy and vignetting in the region of overlap. This aggravated the challenges in compositing the final mosaic.

3.2 Overview

Our goal is to find appropriate geometric transformations between adjacent shots and then stitch them together into one wide-angle view image. The extrinsic parameters of each micro-camera are predetermined by two angular rotations and sensor displacements (Golish et al., 2012), and they should be constant in ideal situations. Given the relative position and rotation and some intrinsic camera parameters, the geometric transformation between adjacent shots can be computed using basic geometry knowledge (Hartley and Zisserman, 2004). Unfortunately, these predetermined parameters suffer from subtle changes over time for reasons such as mechanical or thermal drift and also neighboring images still suffer from parallax error (Wilburn et al., 2005; Kopf et al., 2007; Golish et al., 2012). A minor error in the physical world, such as 1^o error of angular rotation, might result in up to several hundreds pixel shift in image. However, although the predetermined parameters provide us a less accurate geometric transformation for stitching, they still provide a well-accepted global geometric transformation. This inspires our computational pipeline illustrated in Fig. 3.6.

Similar to many standard panorama processing pipelines (Kopf et al., 2007), our pipeline consists of three main components:

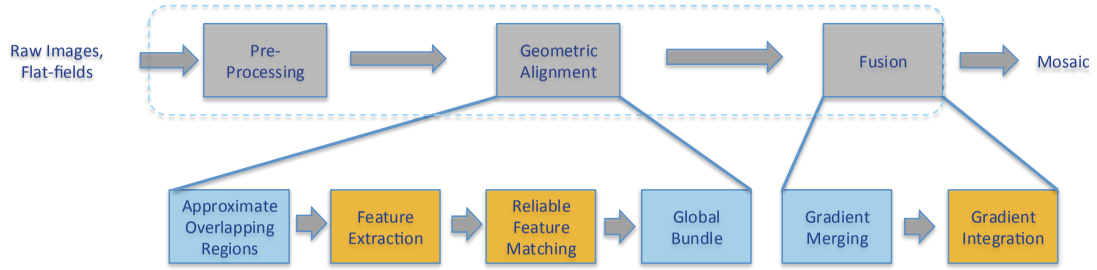


FIGURE 3.6: A simplified diagram of the image stitching process. Computation-intensive modules are highlighted in orange.

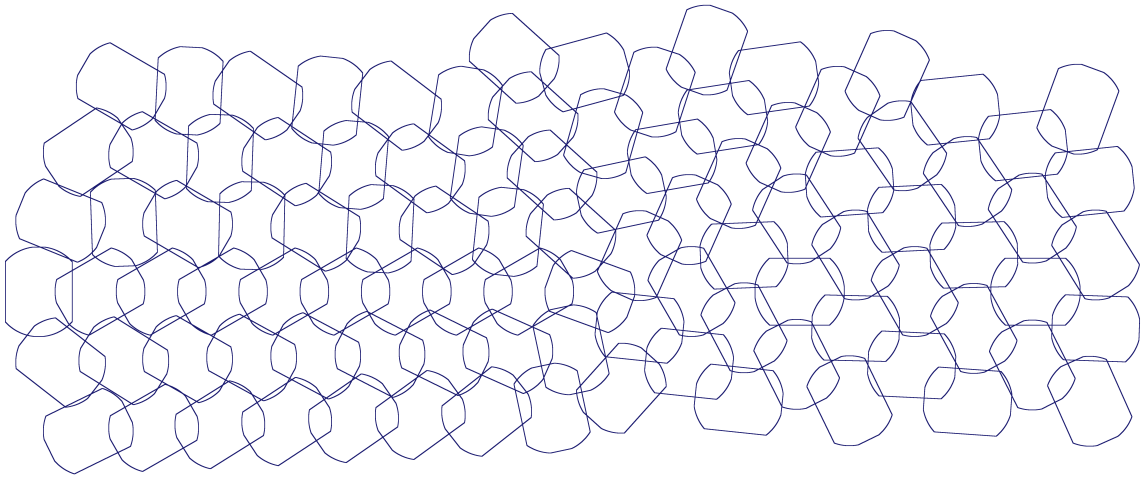


FIGURE 3.7: A diagram of camera FOV distribution.

Preprocessing: The first phase of processing includes demosaicing, de-vignetting, and distortion correction. Unlike many conventional processing pipelines, the while balancing and exposure normalization are not applied here because the output raw images of AWARE-2 are gray scale.

Geometric Alignment: In this stage, we first find the adjacent shots and for each pair of adjacent images, we use feature based alignment technique to find a set of geometrically consistent matches between the images. In conventional camera array systems (Wilburn et al., 2005; Kopf et al., 2007), for each image, they search for the feature matches only in the 8 images known to overlap it. However, AWARE-2 has a different design pattern where each captured image might overlaps with 5-7 neighboring images (See Fig 3.7 for the topology of camera FOV). After that, one way to register all images together is to concatenate pairwise homographies. This would typically cause accumulated errors and a better alternative is to use bundle adjustment (Triggs et al., 2000) to solve for all the camera parameters jointly.

Fusion or Blending: Once the global pose of each image is determined, we can assemble all images together into a composite image. A conventional way is to merge the exposure values of each image. However, the simple division by the shutter speed does not generate exactly matching radiance values in corresponding images due to slight errors in reported shutter speeds (Kopf et al., 2007). Instead, we merge the images in the gradient domain and produce the final composite image by integrating from the gradient domain.

In the following sections, we describe the geometric alignment and fusion steps in detail.

3.3 Pairwise Alignment

The objective of this stage is to align a pair of images. As we mentioned earlier, the major approaches for automatic alignment fall into two categories: direct(pixel)

and feature based. Direct methods are to warp the image relative to each other to maximize the pixel intensity matching. Since direct methods use all of available image data and hence they are typically computational expensive. Besides, direct methods require *brightness constancy* across images which is difficult due to exposure differences. Feature based methods are to find a number of distinct feature points extracted from each images, to establish a geometric correspondence between them, and to then estimate the transformation between images. One advantage of feature based methods is that feature point is relative robust to exposure changes. Today, feature detection and matching are becoming increasing robust. The features are not only respond to conventional "corner" but also to "blob-like" region. Furthermore, many advanced features, such as SIFT, are invariant to location, scale, and rotation, and remarkably robust to change in illumination.

Considering the sparse, geometrically irregular and noisy (S.I.N) property of overlap for micro-camera in AWARE-2, we believe that feature based approach is a better choice and SIFT should be the optimal feature. In order to better support the global-bundle adjustment in the next section, we need to find a number of valid feature matches between adjacent images. The valid feature matches are the feature pairs whose geometric transformation are consistent with the local geometric transformation between adjacent images. Let I_i and I_j be two adjacent images. Following the conventional feature based approaches (Brown and Lowe, 2003; Brown et al., 2005), we first extract and match features between I_i and I_j.

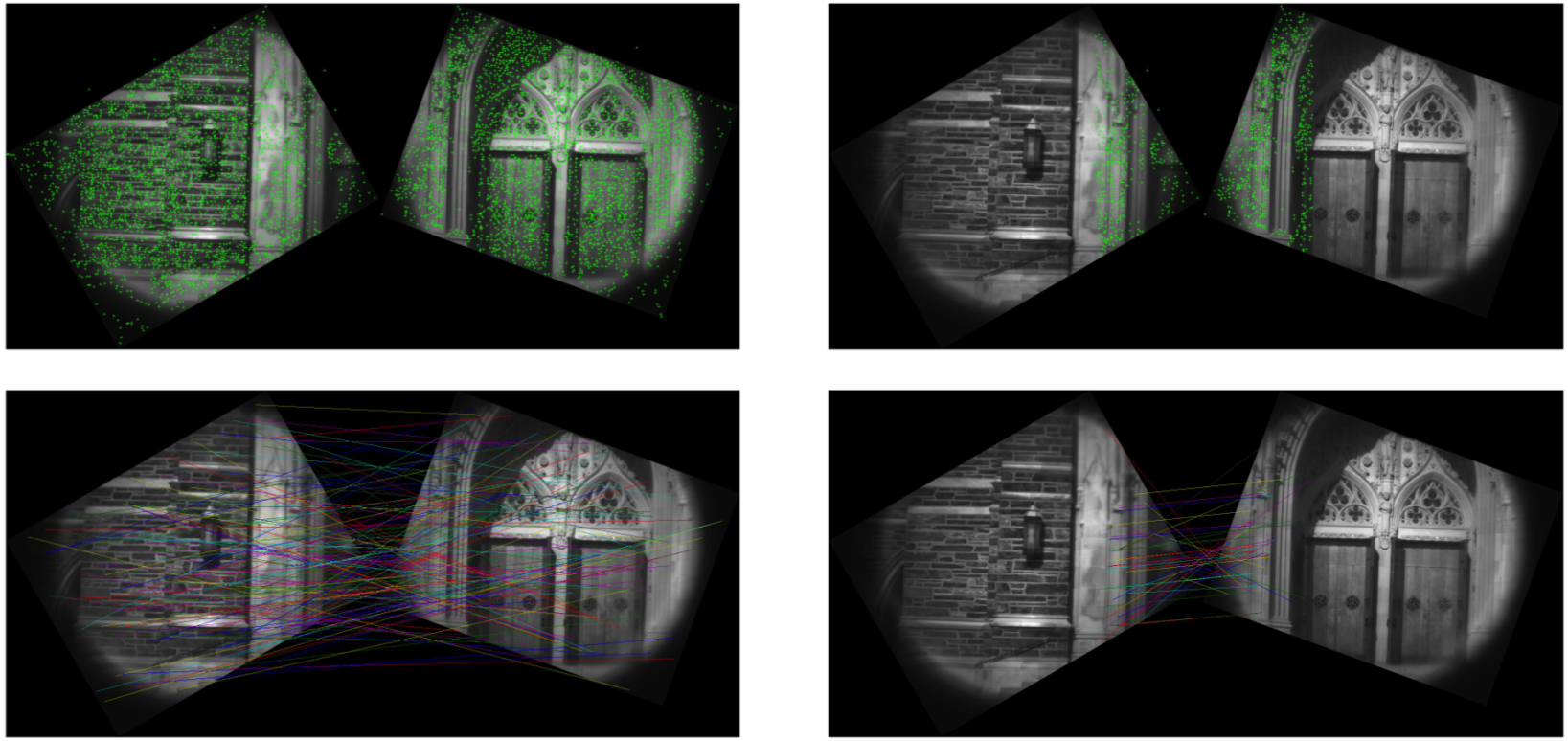


FIGURE 3.8: Top: green points indicates extracted features from whole image region (Left) and from overlap only (Right). Bottom: matched features are connected by lines.

In order to minimize the effects of outlier, we only extract feature in overlapped area between I_i and I_j. The overlapped area can be approximately estimated using the imprecise extrinsic parameters of micro-cameras. In Fig. 3.8, SIFT features are extracted and matched from a pair of example images. As can be seen, the SIFT features are uniformly distributed across whole image domain but most of them are outliers because of sparse overlap between images, thereby most of matched features are false negative (Bottom left in Fig 3.8). On the other hand, if SIFT features are extracted only in overlap between images, most of extracted features are inliers and most of matched features would be true positive (Bottom right in Fig 3.8).

FIGURE 3.9: Left: An example of mismatch between adjacent shots using the matched feature in the feature domain with RANSAC. Right: Result with PG-RANSAC

The extracted features are matched in the features domain, so it is probable that several feature matches are invalid or inconsistent in the geometric aspect. Fig. 3.9 is an example of inconsistent image alignment caused by invalid feature matches. Let $\mathbf{x_k} \leftrightarrow \mathbf{x'_k}$ be the position of kth matched features between I_i and I_j. We consider

them as a valid feature match iff,

$$||(\mathbf{H_{ij}}[\mathbf{x_k^T}, \mathbf{1}]^\mathbf{T}) \times [\mathbf{x_k'^T}, \mathbf{1}]^\mathbf{T}||_2 \leqslant \epsilon \tag{3.1}$$

where $\mathbf{H_{ij}}$ is a 3×3 geometric transformation matrix from I_i and I_j. A standard method to estimate $\mathbf{H_{ij}}$ is to minimize the following function:

$$\mathbf{H_{ij}} = \max_{\mathbf{H}} \sum_k ||(\mathbf{H_{ij}}[\mathbf{x_k^T}, \mathbf{1}]^\mathbf{T}) \times [\mathbf{x_k'^T}, \mathbf{1}]^\mathbf{T}||_2 \tag{3.2}$$

Note that solving Eqn. 3.2 and Eqn. 3.1 is a "chicken and egg" problem: 'ground-truth' model is required in order to classify inlier and outlier in the data set, and for a robust estimation of the model, noise free (or outlier free) data set is preferred. However, if the inlier dominates the feature matches $\{(\mathbf{x_k}, \mathbf{x_k'})\}$, then the 'dead lock' can be resolved by RANSAC algorithm (Fischler and Bolles, 1981). RANSAC fits a homography with four randomly selected feature matches and estimates its quality by counting the number of remaining feature matches that support this transformation. After enough iterations of the above procedure, all the feature matches that fit the best quality homography are inliers with a high probability. RANSAC is very robust to outliers, but typically its performance is highly proportional to the ratio of inlier in the data set.

In order to improve RANSAC's performance even in the presence of a significant amount of outliers, we propose a Placement Geometry-RANSAC (PG-RANSAC) algorithm, which incorporates with the prior knowledge of the geometric transformation derived from the extrinsic parameters of micro-cameras. To this end, we augment the ranking function in the basic RANSAC hypothesis evaluation step to make use of the prior geometric transformation information (See Appendix C for a detailed derivation):

$$rank(\mathbf{H}) = \prod_{i}^{N} \text{rect}\left(\frac{d_i}{2c}\right) g(\mathbf{H}) \tag{3.3}$$

where $\prod_{i}^{N} \text{rect}\left(\frac{d_i}{2c}\right)$ evaluates how well the datum fits with the model $\mathbf{H}$, with d_i is the L_2 distance between the observed position and the predicted position under $\mathbf{H}$, and c is the predetermined threshold in recognizing a datum as the inlier candidate or outlier candidate, and rect(.) is the rect function,

$$\text{rect}(x) = \begin{cases} 1 & \text{when } x \leqslant .5 \\ \text{A small const} & \text{otherwise} \end{cases}$$

The weighting function $g(\mathbf{H})$ penalizes the geometric transformation that deviates much from the expected one, and has the form,

$$g(\mathbf{H}) = \prod_{i}^{N} \frac{1}{1 + e^{-\alpha(\tilde{d}_i - t)}} \cdot \frac{1}{1 + e^{\alpha(\tilde{d}_i - t)}} \tag{3.4}$$

where $\tilde{d}_i$ is the distance between the predicted position under $\mathbf{H}$ and the predicted distance under the expected model. For a single measurement weight $\frac{1}{1 + e^{-\alpha(\tilde{d}_i - t)}} \cdot \frac{1}{1 + e^{\alpha(\tilde{d}_i - t)}}$ is a function that features a plateau in the range $[-t, t]$, and drops sharply at the boundaries at rate dependent on α. Note that if we ignore $g(\mathbf{H})$ or set it to a constant, then maximal $rank(\mathbf{H})$ is equivalent to the most probable hypothesis in the conventional RANSAC algorithm, because:

$$\max \log(rank(\mathbf{H})) \sim \max \sum_{i}^{N} \{d_i \leqslant c\}$$

3.3.1 Global Bundle Adjustment

Let $\mathcal{E} = \{(p,q)|I_p \cap I_q \neq \varnothing \text{ and } q > p\}$ be the index pairs of all adjacent shots. Suppose (p,q) is the lth pair in $\mathcal{E}$, and let $\mathbf{x}_{l,k} \leftrightarrow \mathbf{x}'_{l,k}$ be the kth feature match between I_p and I_q, then all measured data can be organized into a vector as follow:

$$\mathbf{X} = (\mathbf{X_1}^T, \cdots, \mathbf{X_M}^T)^T, L = |\mathcal{E}|$$

$$\mathbf{X_l}^T = (\mathbf{x}_{l,1}^T, \cdots, \mathbf{x}_{l,n}^T, \mathbf{x}_{l,1}'^T, \cdots, \mathbf{x}_{l,n}'^T)^T$$

The goal of global bundle adjustment is to find a set of homography $\{\mathbf{H}_i\}$ that can map each image I_i into a global coordinate system, in which the matched features are mapped to the same position:

$$\mathcal{T}(\mathbf{H}_p[\mathbf{x}_{l,k}^T, \mathbf{1}]^T) = \mathcal{T}(\mathbf{H}_q[\mathbf{x}_{l,k}'^T, \mathbf{1}]^T)) \tag{3.5}$$

$$\mathbf{H}_r = \begin{bmatrix} h_{r,1} & h_{r,2} & h_{r,3} \\ h_{r,4} & h_{r,5} & h_{r,6} \\ h_{r,7} & h_{r,8} & 1 \end{bmatrix}, \quad r \in \{p,q\}$$

where $\mathcal{T}()$ transfers a point from homogenous coordinate to inhomogenous coordinate[1]. Accumulating errors over all measurements, we have:

$$\{\mathbf{H}_i^*\} = \arg\min_{\{\mathbf{H}_i\}} \sum_{(p,q)\in\mathcal{E}} \sum_k ||\mathcal{T}(\mathbf{H}_p[\mathbf{x}_{l,k}^T, \mathbf{1}]^T) - \mathcal{T}(\mathbf{H}_q[\mathbf{x}'^T_{l,k}, \mathbf{1}]^T)||_2^2 \tag{3.6}$$

An alternative way to formulate the optimization is to use true bundle adjustment, i.e., to solve not only for the homography $\{\mathbf{H}_i\}$ but also for the position $\{\mathbf{b}_{l,k}\}$ for each matched feature in the global coordinate system.

$$\{\mathbf{H}_i^*, \mathbf{b_j}\} = \arg\min_{\{\mathbf{H}_i, \mathbf{b_j}\}} \sum_{(p,q)\in\mathcal{E}} \sum_k (||\mathcal{T}(\mathbf{H}_p[\mathbf{x}_{l,k}^T, \mathbf{1}]^T) - \mathbf{b}_{l,k}||_2^2 + ||\mathcal{T}(\mathbf{H}_q[\mathbf{x}'^T_{l,k}, \mathbf{1}]^T) - \mathbf{b}_{l,k}||_2^2) \tag{3.7}$$

[1] $\mathcal{T}(\mathbf{x}) = (x_1/x_3, x_2/x_3)^T, \mathbf{x} = (x_1, x_2, x_3)^T$

The above optimization problem is equivalent to the least square curve fitting the problem as follows:

$$h(\mathbf{P}) = \min_{\mathbf{P}} \sum_{l=1}^{L} ||\mathbf{Y}_l||_2^2 \tag{3.8}$$

$$\text{with } \mathbf{Y}_l = (\mathbf{y}_{l,1}^T, \cdots, \mathbf{y}_{l,n}^T, \mathbf{y'}_{l,1}^T, \cdots, \mathbf{y'}_{l,n}^T)^T$$

$$\mathbf{y}_{l,k} = \mathcal{T}(\mathbf{H}_p[\mathbf{x}_{l,k}^T, \mathbf{1}]^T) - \mathbf{b}_{l,k}$$

$$\mathbf{y'}_{l,k} = \mathcal{T}(\mathbf{H}_q[\mathbf{x'}_{l,k}^T, \mathbf{1}]^T) - \mathbf{b}_{l,k}$$

where $\mathbf{P} = (\mathbf{a}^T, \mathbf{b}^T)^T$ is the concatenated parameter vector: $\mathbf{a} = (\mathbf{h}_1^T, \cdots, \mathbf{h}_M^T)^T$ with $\mathbf{h}_i = \mathbf{Vec}(\mathbf{H}_i)$ and $\mathbf{b} = (\mathbf{b_1}^T, \cdots, \mathbf{b_L}^T)^T$ with $\mathbf{b}_l$ defined as follows:

$$\mathbf{b}_l = (\mathbf{b}_{l,1}^T, \cdots, \mathbf{b}_{l,n}^T)^T$$

This is a non-linear least square problem. It is well known that the problem about $\mathbf{P}$ can be solved using the Leverberg-Marequardt algorithm. Like other numeric optimization algorithms, Leverberg-Marequardt algorithm is an iterative procedure. In each iteration step, the parameter vector $\mathbf{P}$ is replaced by a new estimate $\mathbf{P} + \delta$, where the 'optimal' δ is determined by the equation as follow,

$$(\mathbf{J}^T\mathbf{J} + \lambda \text{ diag}(\mathbf{J}^T\mathbf{J}))\delta = \mathbf{J}\epsilon \tag{3.9}$$

where $\epsilon = \mathbf{0} - \mathbf{Y} = -(\mathbf{Y}_1^T, \cdots, \mathbf{Y}_M^T)^T$ denotes the residential vector using current parameter $\mathbf{P}$ and $\mathbf{J}$ is the Jacobian matrix of this error vector. Note that $\dfrac{\partial \mathbf{Y_i}}{\partial \mathbf{b_j}} = \mathbf{0}$ for $i \neq j$, so $\mathbf{J}$ is a block sparse matrix, as follows:

$$\mathbf{J} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{0} & \mathbf{B}_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{A_L} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B_L} \end{bmatrix}$$

$$\mathbf{A}_i = \frac{\partial \mathbf{Y_i}}{\partial \mathbf{a}}$$

$$\mathbf{B}_i = \frac{\partial \mathbf{Y_i}}{\partial \mathbf{b_i}}$$

For analysis convenience, $\mathbf{J^T J}$ and $\mathbf{J^T}\epsilon$ also can be represented as block matrix,

$$\mathbf{J}^T\mathbf{J} + \lambda \ \mathrm{diag}(\mathbf{J}^T\mathbf{J}) = \begin{bmatrix} \mathbf{U}^*, & \mathbf{W} \\ \mathbf{W^T}, & \mathbf{V}^* \end{bmatrix} \tag{3.10}$$

where

$$\mathbf{U} = \sum_{l=1}^{L} \mathbf{A}_l^T \mathbf{A}_l$$

$$\mathbf{U}^* = \mathbf{U} + \lambda \ \mathrm{diag}(\mathbf{U})$$

$$\mathbf{V} = \mathrm{diag}(\mathbf{B}_1^T\mathbf{B}_1, \cdots, \mathbf{B}_L^T\mathbf{B}_L)$$

$$\mathbf{V}^* = \mathbf{V} + \lambda \ \mathrm{diag}(\mathbf{V})$$

$$\mathbf{W} = \left[\mathbf{W}_1, \cdots, \mathbf{W}_L\right]$$

$$\mathbf{W}_l = \mathbf{A}_l^T\mathbf{B}_l$$

and

$$\mathbf{J}^T\epsilon = \mathbf{J}^T\left[\epsilon_1^T, \cdots, \epsilon_M^T\right]^T = \begin{pmatrix} \epsilon_a \\ \epsilon_b \end{pmatrix} \tag{3.11}$$

with

$$\epsilon_{\mathbf{a}} = \sum_{l=1}^{L} \epsilon_{a,l}$$

$$\epsilon_{a,l} = \mathbf{A}_l^T \epsilon_l$$

$$\epsilon_b = (\epsilon_{b,1}^T, \cdots, \epsilon_{b,L}^T)^T$$

$$\epsilon_{b,l} = (\mathbf{B}_l^T \epsilon_l)$$

Now multiplying both sides of Eqn. 3.9 with $\begin{bmatrix} \mathbf{I} & -\mathbf{W}\mathbf{V}^{*-\mathbf{1}} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$, a *triangular* linear system is formed:

$$\begin{bmatrix} \mathbf{U}^* - \mathbf{W}\mathbf{V}^{*-\mathbf{1}}\mathbf{W}^{\mathbf{T}} & \mathbf{0} \\ \mathbf{W}^{\mathbf{T}} & \mathbf{V}_* \end{bmatrix} \begin{pmatrix} \delta_a \\ \delta_b \end{pmatrix} = \begin{pmatrix} \epsilon_{\mathbf{a}} - \mathbf{W}\mathbf{V}^{*-\mathbf{1}}\epsilon_b \\ \epsilon_b \end{pmatrix} \tag{3.12}$$

The above linear system can be solved by back-substitution algorithm and a naïve implementation has a complexity of $\mathcal{O}(M^2N)$, where M is the total number of images (or cameras in our case) and N is the total number of matched features. In Appendix D, we discuss the sparseness of the linear system and reduce the algorithm complexity to $\mathcal{O}(M + N)$.

3.4 Image Blending

Following the bundle adjustment process, every pixel can be projected onto a specific location of the mosaic canvas. Ideally, the intensity of pixels projected to the same location should be the same, but in reality this is not the case. Due to changes in aperture, exposure time and vignetting of different micro-cameras, adjacent shots might have apparently photometric variation and a naïve blending always introduces visible artificial edges.

An intuitional approach that can deal with the visible artifacts is to perform a weighted combination of the pixel intensity from different sources. The weighting coefficients are spatially varying and depend on the distance to image boundary, such as alpha blending, or the distance to optimal seam where the intensity transition between image are minimal (Efros and Freeman, 2001). A more advanced approach is to combine them in multiple bands - *Laplacian pyramid*, in order to retain sharp enough transitions to prevent blurring (Burt et al., 1983; Brown and Lowe, 2003).

An alternative approach is to blend images in the gradient domain. Gradient domain processing has a long history in image editing, such as high dynamic range tone-mapping (Fattal et al., 2002), seamless object insertion (Pérez et al., 2003), image enhancement (Bhat et al., 2008) and image stitching (Levin et al., 2006). Several advantages for the gradient domain processing: First, high frequency information can be preserved in the gradient domain. Second, image gradient are invariant to sensor bias. Moreover, the combination of gradient produces a visually smoothing intensity transition in the overlapped areas between different images.

We adopt the approach of blending images in the gradient domain. Let F indicate the final expected mosaic,

$$\nabla F(\mathbf{x}) = \sum_i w_i(\mathcal{T}(\mathbf{H}_i^{-1}\mathbf{x}))I_i(\mathcal{T}(\mathbf{H}_i^{-1}\mathbf{x})) \tag{3.13}$$

where $w_i(\mathbf{x})$ is a weighing function that we set to be proportional to the pre-calibrated flat-field measurement, which reflects the apparent gain and dark current of each pixel. $w_i(\mathbf{x})$ is zero for pixel $\mathbf{x}$ outside the image domain Ω_i of I_i. H_i is the homography maps from the local image coordinate of I_i to the global image coordinate in F.

We employ *Neumann boundary conditions* for the pixels outside the image domain of F. With some discretizations, Eqn. 3.13 can be converted into a sparse linear system that can be solved using existing numeric techniques, such as multi-grid

and preconditioned conjugate gradient. An alternative and more efficient method is to approximate the integration operation. We adopt the convolution pyramid scheme (Farbman et al., 2011) to compute the approximated optimal solution for Eqn. 3.13. Note that the solution of Eqn. 3.13 is not unique because any optimal solution with an additional constant is still the optimal solution. We enforce the intensity mean of F is the same as that of $\{I_i\}$ to resolve the ambiguity (See Fig. 3.10)

FIGURE 3.10: Top: Result with blending in the intensity domain. Bottom: Result after blending in the gradient domain and integration.

3.5 Results and Additional Remarks

Our pipeline was protptyped in Matlab and an optimized version with utilization of multi-core and GPU was published in Iliopoulos et al. (2013). In the aspect of computation, current pipeline takes 50s in processing a mosaic of 100M pixels, while a naïve serial implementation costs up to 3.5 hours.

As the baseline to compare against, we conducted experiments with AWARE-2 compositing pipeline (Golish et al., 2012). The experimental results are obtained with our pipeline on real data acquired with AWARE-2. Figure 3.12 and 3.11 show our results sensibly better than baseline method. In our result, (right in Figure 3.12 and 3.11), the face of person and the tree branch are more visual artifacts free, while apparent ghost appears in the results produced by the baseline method (left in Figure 3.12 and 3.11). As we mentioned in previous section, our gradient is relative robust to bias gain and exposure change, so the gradient based fusion could generate more pleasant results on the overlapped area.

FIGURE 3.11: Snapshot mosaics of a live scene, captured with the AWARE-2 camera prototype at the Hudson Building, Duke University. *Left:* Results produced by the AWARE-2 compositing pipeline (Golish et al., 2012), where tone mapping has been applied. *Right:* Results produced automatically by our method, without tone mapping.

FIGURE 3.12: Snapshot mosaics of a live scene, captured with the AWARE-2 camera prototype at the ICCP, Seattle, 2012 (bottom floor and architectural surroundings not shown). *Left-top, Left-bottom:* Results produced by the AWARE-2 compositing pipeline (Golish et al., 2012), where tone mapping has been applied. *Right-top, Right-bottom:* Results produced automatically by our method, without tone mapping. *Top row*: The displayed scene spans the fields of view of approximately 25 micro-cameras. *Bottom row:* Detail, zoomed in within the marked windows in the top-row images.

4

Conclusion

While the acquisition of an image stack is enabled by advanced technologies and design, digital processing of the stack images is to overcome the remaining limitation and render synthesized image that is better, in one or more than one characteristics, than any of the individual raw image. This work is concerned with the improvement in more than one rendering characteristics. I have studied certain common issues in digital stack photography with two different and important applications, one is high-dynamic range (HDR) composition and the other is panorama stitching. In the HDR imaging, I am concerned with the photometric effect and the effect of motion (rigid or non-rigid) or other dynamic changes. In the panorama imaging, I am concerned with the sparse, irregular and noise conditions, among other systematic and dynamic changes.

Previous work with HDR imaging assumes commonly stationary scenes. We were among the first to compose high quality HDR image from exposure stacks in the presence of both camera motion and scene changes (Hu et al., 2012) . The idea behind our methods is to model both geometric and photometric changes in the exposure stacks, and to use EM-like algorithm to solve both changes iteratively. I

choose the patch-based model for geometric alignment which allows to deal with large scale non-rigid motion in an efficient way and the photometric change is modeled as close as physical model for exposure change in real cameras. Even in the presence of clipped pixels in the reference, our method can be extended with a patch-based texture synthesis model to produce pleasant results (Hu et al., 2013). Moreover, the estimation of geometric alignment and photometric transformation are relative independent in our algorithm. Therefore, the proposed algorithms could be extended to other applications by pursuing different forms of photometric change, such as focal-stacking and flash/non-flash stack. The price we pay behind these better and robust models are more memory and computation resources. Fortunately, these kinds of challenges will be resolved with the evolution of electronic storage and computing power.

Conventional panorama techniques capture a series of images with $10\% - 30\%$ overlap across different angle of view in sequence , and register and stitch them incrementally using feature based alignment. The sequentially captured images typically introduce ghost caused by moving objects in non-stationary scenes. The camera array system, such as AWARE-2, allows to capture all image stacks simultaneously, which can be treated as if shot in succession by a single camera capturing an effectively stationary scene. However, existing approaches are challenged by the data that are noisy, or lack of feature, over sparse $(5\% - 10\%)$ overlap of irregular geometry. In addition, one could not expect the calibration of a system with many cameras as that of a single camera. The complexity of the relative relations between the distributed cameras grows exponentially with the number of camera. The solution presented in this thesis is to follow the approximate the geometry by design, to allow imprecision in calibration, and to adapt to the deviations and changes by incorporating in image alignment the systematic geometric information and data-specific features and statistics.

It remains a great challenge to process with multiple stacks, say, to compose a HDR panorama. Multiple stacks are not necessarily independent of each other. For instance, the dynamic range of the scene captured by each component image for panorama composition is not higher, and perhaps much lower, than that of the entire scene of interest. The high dynamic range of each component may be better determined not only by its own exposure stack but also by those stacks of its neighbors. The challenge is in the range mapping for the panoramic one.

Digital stack processing will continue advancing in many fronts, bridging the advance in data acquisition, the rendering capabilities and the growing demand for useful information and appealing characteristics. For example, early and big camera array systems (Wilburn et al., 2005; Golish et al., 2012) will be replaced by smaller, cheaper, smarter camera-array systems, which will be equipped and enriched by other types of sensors and will be introduced into household, vehicle-hold, and hand-hold devices, beyond research laboratories. Recently, Venkataraman et al. (2013) presented an ultra-thin high performance monolithic 4×4 camera array, that captures light fields and synthesizes high resolution image along with a range image. In computational photography, image stabilization and panorama stitching have benefited from gyroscope and accelerometer. The multiple modality makes many practical computer vision problems better posed and easier to solve. Multi-modal sensor systems have appeared in many practical applications. For instance, Google's self-driving car combines data from multiple sensors, such as 64-beam laser, video camera and radar, to generate a detailed 3D map in real time (Thrun, 2011). Meanwhile, rendering technologies and techniques will change accordingly, such as 3D portrait and 3D panorama. Furthermore, it is not hard to imagine that mobile devices and mobile computing network will reshape image acquisition, processing and rendering in the near future.

Appendix A

Brightness Transfer Function

Using the conventional radiometric model, the measured intensity $I(\mathbf{x})(\mathbf{x} \in \Omega)$ is related with the scene irradiance $E(\mathbf{x})$ by the camera response function f as follows:

$$I(\mathbf{x}) = f\left(E(\mathbf{x})\right) \tag{A.1}$$

For a pair of aligned images $\{R, S\}$ with an exposure ratio of k, we have:

$$\frac{f^{-1}\left(S(\mathbf{x})\right)}{f^{-1}\left(R(\mathbf{x})\right)} = k \tag{A.2}$$

The above derivation is based on two assumptions: First, the scene is static and is *radiance constancy*. Second, the camera response function f is invertible. Moreover, we have:

$$S(\mathbf{x}) = f\left(kf^{-1}(r)\right) = \tau(r) \tag{A.3}$$
$$r = R(\mathbf{x})$$

It is apparent that $\tau(r)$ is a smoothly monotonic function as the camera response function f is a smoothly monotonic function.

Appendix B

Brightness Transfer Function Approximation

Let $\tau(r)$ be piecewise Hermite cubic splines parameterized by $\{p_k, m_k\}$ for $k = 1, \cdots, n$, and, therefore, for the kth interval $[p_k, p_{k+1}]$, the Hermite cubic spline is:

$$(2t^3 - 3t^2 + 1)p_k + (t^3 - 2t^2 + t)m_k + (-2t^3 + 3t^2)p_{k+1} + (t^3 - t^2)m_{k+1} \qquad \text{(B.1)}$$

where $t \in (0, 1)$ is locally normalized coordinate. Let $R(\mathbf{x})$ be in the ith term of $\sum_{\mathbf{x}} \psi\left(\tau(R(\mathbf{x}) - S_{\mathbf{u}}(\mathbf{x}))\right)$ in Eq. 2.7 and suppose $r = R(\mathbf{x})$ in kth interval of $\tau(r)$, then:

$$\begin{aligned} \tau(R(\mathbf{x})) &= (2\bar{r}^3 - 3\bar{r}^2 + 1)p_k + (\bar{r}^3 - 2\bar{r}^2 + \bar{r})m_k \\ &\quad + (-2\bar{r}^3 + 3\bar{r}^2)p_{k+1} + (\bar{r}^3 - \bar{r}^2)m_{k+1} \\ &= \mathbf{a_i p} \end{aligned} \qquad \text{(B.2)}$$

where

$$\mathbf{a_i} = \left[\mathbf{0}_{2k-2}, (2\bar{r}^3 - 3\bar{r}^2 + 1), (\bar{r}^3 - 2\bar{r}^2 + \bar{r}), (-2\bar{r}^3 + 3\bar{r}^2), (\bar{r}^3 - \bar{r}^2), \mathbf{0}_{2(n-k)-2}\right]$$

$$\mathbf{p} = [p_0, m_0, p_1, m_1, \cdots, p_n, m_n]^T$$

$$\mathbf{0}_t = \underbrace{[0, \cdots, 0]}_{t \text{ zeros}}$$

$$\bar{r} = (r - p_k)/(p_{k+1} - p_k)$$

and let

$$b_i = S_{\mathbf{u}}(\mathbf{x}) \tag{B.3}$$

so in Eq. 2.7:

$$\sum_{\mathbf{x}} \psi(\tau(R(\mathbf{x}) - S_{\mathbf{u}}(\mathbf{x}))) = \sum_i \psi(\mathbf{a_i p} - b_i) \approx \psi(\mathbf{Ap} - \mathbf{b}) \tag{B.4}$$

Let $\{r_i\}$ for $i = 1, \cdots, N$ be the domain of $\tau(r)$. It is always a finite discrete in practice. As in Eq. B.2, we can build a matrix $\mathbf{M}$ as follows:

$$\mathbf{Mp} = [\tau(r_1), \cdots, \tau(r_N)]^T \tag{B.5}$$

Let $\mathbf{D}$ be a $(N-1) \times N$ differential operator, and the hard monotonicity constraint in Eq. 2.7 can be approximated as follows:

$$\tau'(r) \geqslant 0 \Leftrightarrow \mathbf{Tp} = \mathbf{DMp} \geqslant 0 \tag{B.6}$$

Appendix C

Placement Geometric RANSAC

Let $\mathcal{D}$ be the measured data set. RANSAC assesses a hypothesis model $\mathbf{H}$ by counting how many of the datum in $\mathcal{D}$ are well-fitted as 'inliers', but a more considered method is to score using the posterior probability $\Pr(\mathbf{H}|S)$. The posterior probability is not directly measureable, but by Bayes' Theorem:

$$\Pr(\mathbf{H}|\mathcal{D}) = \Pr(\mathcal{D}|\mathbf{H}) \Pr(\mathbf{H}) / \Pr(\mathcal{D}) \tag{C.1}$$

as the prior probability of the observation, $\Pr(\mathcal{D})$ is a constant, so:

$$\max \Pr(\mathbf{H}|\mathcal{D}) \sim \max \Pr(\mathcal{D}|\mathbf{H}) \Pr(\mathbf{H}) \tag{C.2}$$

If the prior probability of $\mathbf{H}$ is assumed uniform, the problem reduces to conventional RANSAC as follows:

$$\max \Pr(\mathbf{H}|\mathcal{D}) \sim \max \Pr(\mathcal{D}|\mathbf{H}) \tag{C.3}$$

However, for more general cases, a prior probability $\Pr(\mathbf{H})$ should be considered.

Appendix D

Global Bundle Adjustment

Eqn. 3.12 can be solved using back-substitution algorithm as follows:

$$\left[\sum_{l=1}^{L}\mathbf{A}_l^T\mathbf{A}_l + \lambda\ \text{diag}\left(\sum_{l=1}^{L}\mathbf{A}_l^T\mathbf{A}_l\right) - \sum_{l=1}^{L}\mathbf{W}_l\mathbf{V}_l^{*-1}\mathbf{W}_l^T\right]\delta_a = \epsilon_a - \sum_{l=1}^{L}\mathbf{W}_l\mathbf{V}_l^{*-1}\epsilon_{b,l} \tag{D.1}$$

$$\delta_{b,l} = \mathbf{V}_l^{*-1}(\epsilon_{b,l} - \mathbf{W}_l^T\delta_a) \tag{D.2}$$

In the remainder of this section, we will show how to reduce the algorithm complexity from $\mathcal{O}(M^2N)$ to $\mathcal{O}(M+N)$, where M is number of images and N is the total number of matched features. The idea is to look at each term in the above equations as block matrix, and to explore the sparseness of the block matrices to avoid the redundant computations with zero block matrices.

Lemma 1. *Let* $\mathbf{Y}_l = \left[\mathbf{Y}_l^{1T}, \mathbf{Y}_l^{2T}\right]^T$*, with*

$$\mathbf{Y}_l^1 = \left[\mathbf{y}_{l,1}^T, \cdots, \mathbf{y}_{l,n}^T\right]^T$$

$$\mathbf{Y}_l^2 = \left[\mathbf{y}'^T_{l,1}, \cdots, \mathbf{y}'^T_{l,n}\right]^T$$

then we have,

$$\frac{\partial \mathbf{y}_{l,k}}{\partial \mathbf{b}_{l,i}} = \frac{\partial \mathcal{T}(\mathbf{H}_p[\mathbf{x}_{l,k}^T, \mathbf{1}]^T) - \mathbf{b}_{l,k}}{\partial \mathbf{b}_{l,i}} = \begin{cases} \mathbf{0} & \text{if } i \neq k \\ -\mathbf{I} & \text{otherwise} \end{cases}$$

$$\frac{\partial \mathbf{y}'_{l,k}}{\partial \mathbf{b}_{l,i}} = \frac{\partial \mathcal{T}(\mathbf{H}_q[\mathbf{x}'^T_{l,k}, \mathbf{1}]^T) - \mathbf{b}_{l,k}}{\partial \mathbf{b}_{l,i}} = \begin{cases} \mathbf{0} & \text{if } i \neq k \\ -\mathbf{I} & \text{otherwise} \end{cases}$$

$$\frac{\partial \mathbf{y}_{l,k}}{\partial \mathbf{h}_s} = \frac{\partial \mathcal{T}(\mathbf{H}_p[\mathbf{x}_{l,k}^T, \mathbf{1}]^T) - \mathbf{b}_{l,k}}{\partial \mathbf{h}_s} = \mathbf{0} \text{ if } s \neq p$$

$$\frac{\partial \mathbf{y}'_{l,k}}{\partial \mathbf{h}_s} = \frac{\partial \mathcal{T}(\mathbf{H}_q[\mathbf{x}'^T_{l,k}, \mathbf{1}]^T) - \mathbf{b}_{l,k}}{\partial \mathbf{h}_s} = \mathbf{0} \text{ if } s \neq q$$

According to the Lemma 1, $\mathbf{A}_l$ is a sparse block matrix as follows:

$$\mathbf{A}_l = \frac{\partial \mathbf{Y}_l}{\partial \mathbf{a}} = \left[\frac{\partial \mathbf{Y}_l^r}{\partial \mathbf{h}_s}\right]_{r,s}$$

$$r \in \{1, 2\}$$

$$s \in \{1, \cdots, N\}$$

with

$$\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_s} = \left[\frac{\partial \mathbf{y}_{l,1}}{\partial \mathbf{h}_s}^T, \cdots; \frac{\partial \mathbf{y}_{l,n}}{\partial \mathbf{h}_s}^T\right]^T = \mathbf{0} \text{ if } s \neq p$$

$$\frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_s} = \left[\frac{\partial \mathbf{y}'_{l,1}}{\partial \mathbf{h}_s}^T, \cdots; \frac{\partial \mathbf{y}'_{l,n}}{\partial \mathbf{h}_s}^T\right]^T = \mathbf{0} \text{ if } s \neq q$$

Fig. D.1 provides a visual demonstration of matrix $\mathbf{A}_l$. The red blocks are zero block matrices and the white blocks are nonzero block matrices. Furthermore, it

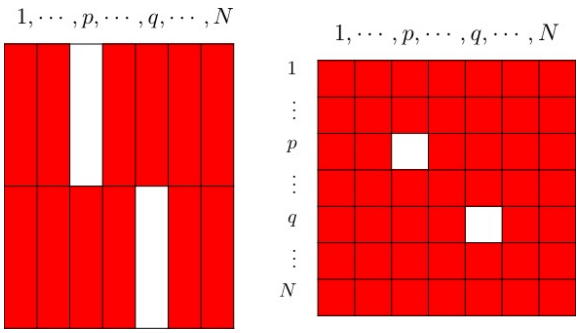


FIGURE D.1: Visualization of sparse block matrix $\mathbf{A}_l$ (Left) and $\mathbf{A}_l^T\mathbf{A}_l$ (Right). Red block is zero sub-matrix, while white block is non-zero.

is easy to find that $\mathbf{A}_{ll} = \mathbf{A}_l^T\mathbf{A}_l$ is also a sparse block matrix. Let $\mathbf{A}_{ll,rs}$ be the sub-block matrix at rth row and sth col of $\mathbf{A}_{ll}$, then,

$$\mathbf{A}_{ll,rs} = \begin{cases} \left(\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p}\right)^T \frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p} & \text{if } r = p, s = p \\ \left(\frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_q}\right)^T \frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_q} & \text{if } r = q, s = q \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{D.3}$$

According to the Lemma 1, $\mathbf{B}_l$ can be derived as a sparse block matrix too,

$$\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{b}_l} = \left[\frac{\partial \mathbf{y}_{l,i}}{\partial \mathbf{b}_{l,j}}\right]_{i,j} = -\mathbf{I}$$

$$\frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{b}_l} = \left[\frac{\partial \mathbf{y'}_{l,i}}{\partial \mathbf{b}_{l,j}}\right]_{i,j} = -\mathbf{I}$$

$$\mathbf{B}_l = \frac{\partial \mathbf{Y}_l}{\partial \mathbf{b}_l} = \begin{bmatrix} \frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{b}_l} \\ \\ \frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{b}_l} \end{bmatrix} = -\begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix}$$

therefore,

$$\mathbf{V}_l = \mathbf{B}_l^T \mathbf{B}_l = 2\mathbf{I}$$

$$\mathbf{V}_l^* = (2\lambda + 2)\mathbf{I}$$

$$\mathbf{W}_l = \mathbf{A}_l^T \mathbf{B}_l = -\left[\left(\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_r} + \frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_r}\right)^T\right]_r$$

$$\mathbf{W}_l \mathbf{W}_l^T = \left[\left(\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_r} + \frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_r}\right)^T \left(\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_s} + \frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_s}\right)\right]_{r,s}$$

Again, Lemma 1 tells us that $\mathbf{W}_l$ and $\mathbf{W}_l \mathbf{W}_l^T$ are block sparse matrices (See Fig. D.2). Let $\mathbf{W}_{l,r}$ denotes the rth row block in $\mathbf{W}_l$ and $\mathbf{W}_{ll,rs}$ be the sub-block matrix at rth row, sth col of $\mathbf{W}_l \mathbf{W}_l^T$, then,

$$\mathbf{W}_{l,r} = \begin{cases} -\left(\dfrac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p}\right)^T & \text{if } r = p \\ -\left(\dfrac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_q}\right)^T & \text{if } r = q \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\mathbf{W}_{ll,rs} = \begin{cases} \left(\dfrac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p}\right)^T \dfrac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p} & \text{if } r = p, s = p \\ \left(\dfrac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_q}\right)^T \dfrac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_q} & \text{if } r = q, s = q \\ \left(\dfrac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p}\right)^T \dfrac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_q} & \text{if } r = p, s = q \\ \left(\dfrac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_q}\right)^T \dfrac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p} & \text{if } r = q, s = p \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{D.4}$$

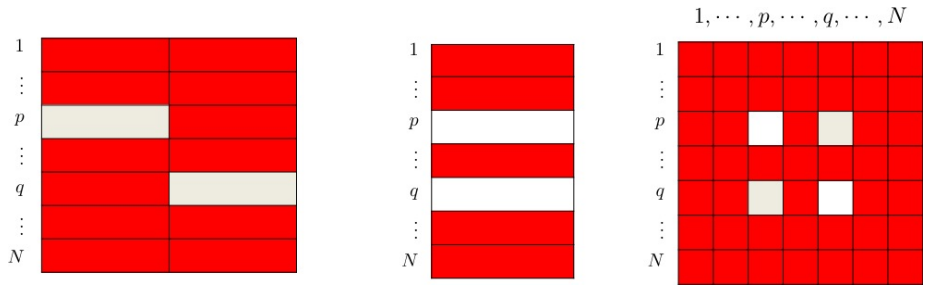


FIGURE D.2: Visualization of sparse block matrix $\mathbf{A}_l^T$ (Left), $\mathbf{W}_l$ (Middle) and $\mathbf{W}_l\mathbf{W}_l^T$ (Right). Red block is a zero sub-matrix, while white block is a non-zero sub-matrix.

Let $\epsilon_l = [\epsilon_l^{1T}, \epsilon_l^{2T}]^T$ with ϵ_l^1 and ϵ_l^2 corresponding to the current error vector for $\mathbf{Y}_l^1$ and $\mathbf{Y}_l^2$, respectively.

$$\epsilon_{a,l} = \mathbf{A}_l^T \epsilon_l = \left[\left(\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_r} \right)^T \epsilon_l^1 + \left(\frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_r} \right)^T \epsilon_l^2 \right]_r$$

Lemma 1 tells us that $\epsilon_{a,l} = \mathbf{A}_l^T \epsilon_l$ is a sparse vector as shown in Fig. D.3, and more specifically:

$$\epsilon_{a,l}(r) = \begin{cases} \left(\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p} \right)^T \epsilon_l^1 & \text{if } r = p \\ \left(\frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_q} \right)^T \epsilon_l^2 & \text{if } r = q \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{D.5}$$

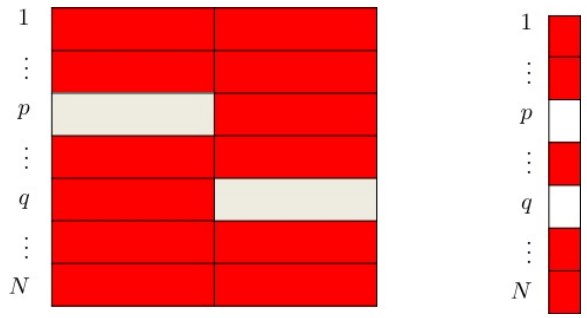


FIGURE D.3: Visualization of sparse block matrix $\mathbf{A}_l^T$ (Left) and $\mathbf{A}_l^T \epsilon_l$ (Right). Red block is zero sub-matrix, while white block is a non-zero sub-matrix.

Similar conclusion for ϵ_b and $\mathbf{W}_l \mathbf{V}_l^{*-1} \epsilon_{b,l}$:

$$\epsilon_{b,l} = \mathbf{B}_l^T \epsilon_l = -(\epsilon_l^1 + \epsilon_l^2)$$

$$\mathbf{W}_l \mathbf{V}_l^{*-1} \epsilon_{b,l} = \frac{-1}{2\lambda + 2} \mathbf{A}_l^T \mathbf{B}_l (\epsilon_l^1 + \epsilon_l^2) = \frac{1}{2\lambda + 2} \mathbf{A}_l^T \left[(\epsilon_l^1 + \epsilon_l^2)^T, (\epsilon_l^1 + \epsilon_l^2)^T \right]^T$$

Let $\epsilon'_{b,l} = \mathbf{W}_l \mathbf{V}_l^{*-1} \epsilon_{b,l}$, then,

$$\epsilon'_{b,l}(r) = \begin{cases} \frac{1}{2\lambda + 2} \left(\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p} \right)^T (\epsilon_l^1 + \epsilon_l^2) & \text{if } r = p \\ \frac{1}{2\lambda + 2} \left(\frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_q} \right)^T (\epsilon_l^1 + \epsilon_l^2) & \text{if } r = q \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{D.6}$$

Let $\delta_a = [\delta_{h_1}^T, \cdots, \delta_{h_M}^T]^T$ with δ_{h_l} corresponding to the increment value for the parameter $\mathbf{h}_l$,

$$\mathbf{W}_l^T = \mathbf{B}_l^T \mathbf{A}_l = -\left[\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_s} + \frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_s} \right]_s$$

$$\mathbf{W}_l^T \delta_a = -\sum_s \left(\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_s} + \frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_s} \right) \delta_{h_s} = -\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p} \delta_{h_p} - \frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h}_p} \delta_{h_q} \tag{D.7}$$

Now each term in Eqn. D.1 can be computed using the block matrices from Eqn. D.3, D.4, D.5 , D.6 , D.7. Because of the spareness of these matrices, they can be computed in a very cheap way. Besides, the basic computation block $\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h_p}}$ (also $\frac{\partial \mathbf{Y}_l^2}{\partial \mathbf{h_q}}$) can be computed in a very efficient way as follows:

$$\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p} = \left[\frac{\partial \mathbf{y}_{l,1}}{\partial \mathbf{h}_p}^T, \cdots ; \frac{\partial \mathbf{y}_{l,n}}{\partial \mathbf{h}_p}^T \right]^T$$

$$\frac{\partial \mathbf{y}_{l,k}}{\partial \mathbf{h_p}} = \begin{bmatrix} x_k & y_k & 1 & 0 & 0 & 0 & \frac{-x_k^2 h_1 - x_k y_k h_2 - x_k h_3}{x_k h_7 + y_k h_8 + 1} & \frac{-x_k y_k h_1 - y_k^2 h_2 - y_k h_3}{x_k h_7 + y_k h_8 + 1} \\ 0 & 0 & 0 & x_k & y_k & 1 & \frac{-x_k^2 h_4 - x_k y_k h_5 - x_k h_6}{x_k h_7 + y_k h_8 + 1} & \frac{-x_k y_k h_4 - y_k^2 h_5 - y_k h_6}{x_k h_7 + y_k h_8 + 1} \end{bmatrix} / \mathbf{D_l}$$

$$\mathbf{D_l} = x_k h_7 + y_k h_8 + 1$$

where $\mathbf{y}_{l,k} = (x_k, y_k)^T$. Now let us define some intermediate variables:

$$\mathbf{A} = \begin{bmatrix} x_1 & y_1 & 1 \\ 0 & 0 & 0 \\ x_2 & y_2 & 1 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \\ 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ x_1 & y_1 & 1 \\ 0 & 0 & 0 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ x_n & y_n & 1 \end{bmatrix}$$

$$\mathbf{D} = \mathbf{A} + \mathbf{B} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \\ x_n & y_n & 1 \end{bmatrix}$$

$$\mathbf{h}_p = \begin{bmatrix} \mathbf{h}_p^1 \\ - \\ \mathbf{h}_p^2 \\ - \\ \mathbf{h}_p^3 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ 1 \end{bmatrix}$$

$$\mathbf{C}_1 = \begin{bmatrix} -x_1^2 & -x_1y_1 & -x_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -x_1^2 & -x_1y_1 & -x_1 \\ -x_2^2 & -x_2y_2 & -x_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -x_2^2 & -x_2y_2 & -x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_n^2 & -x_ny_n & -x_n & 0 & 0 & 0 \\ 0 & 0 & 0 & -x_n^2 & -x_ny_n & -x_n \end{bmatrix}$$

$$\mathbf{C}_2 = \begin{bmatrix} -x_1y_1 & -y_1^2 & -y_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -x_1y_1 & -y_1^2 & -y_1 \\ -x_2y_2 & -y_2^2 & -y_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -x_2y_2 & -y_2^2 & -y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_ny_n & -y_n^2 & -y_n & 0 & 0 & 0 \\ 0 & 0 & 0 & -x_ny_n & -y_n^2 & -y_n \end{bmatrix}$$

Now $\dfrac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p}$ can be computed with some basic matrix operations:

$$\frac{\partial \mathbf{Y}_l^1}{\partial \mathbf{h}_p} = \left[\; \mathbf{A} \circ [\hat{\mathbf{D}}, \hat{\mathbf{D}}, \hat{\mathbf{D}}] \;\middle|\; \mathbf{B} \circ [\hat{\mathbf{D}}, \hat{\mathbf{D}}, \hat{\mathbf{D}}] \;\middle|\; \hat{\mathbf{C}}_\mathbf{1} \circ \hat{\mathbf{D}} \circ \hat{\mathbf{D}} \;\middle|\; \hat{\mathbf{C}}_\mathbf{2} \circ \hat{\mathbf{D}} \circ \hat{\mathbf{D}} \; \right] \tag{D.8}$$

$$\hat{\mathbf{C}}_\mathbf{1} = \mathbf{C}_1 \left[\mathbf{h}_p^{1^T}, \mathbf{h}_p^{2^T} \right]^T$$

$$\hat{\mathbf{C}}_\mathbf{2} = \mathbf{C}_2 \left[\mathbf{h}_p^{1^T}, \mathbf{h}_p^{2^T} \right]^T$$

$$\hat{\mathbf{D}} = 1./(\mathbf{D}\mathbf{h}_p^3)$$

where $\circ$ is *Hadamard* product and ./ is element-wise division.

Bibliography

Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., and Cohen, M. (2004), "Interactive digital photomontage," *ACM Transactions on Graphic*, 23, 294–302.

Akyüz, A. O. and Reinhard, E. (2007), "Noise reduction in high dynamic range imaging," *Visual Communication and Image Representation*, 18, 366–376.

Baker, S. and Matthews, I. (2004), "Lucas-Kanade 20 Years On: A Unifying Framework," *International Journal of Computer Vision*, 56, 221–255.

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R. (2010), "A Database and Evaluation Methodology for Optical Flow," *International Journal of Computer Vision*, 92, 1–31.

Balme, M. R., Pathare, A., Metzger, S. M., Towner, M. C., Lewis, S. R., Spiga, A., Fenton, L. K., Renno, N. O., Elliott, H. M., Saca, F. A., Michaels, T., Russell, P., and Verdasca, J. (2012), "Field measurements of horizontal forward motion velocities of terrestrial dust devils: Towards a proxy for ambient winds on Mars and Earth," *ICARUS*, 221, 632–645.

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009), "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing," *ACM Transactions on Graphic*, 28, 24:1–24:11.

Barnes, C., Shechtman, E., Goldman, D. B., and Finkelstein, A. (2010), "The Generalized PatchMatch Correspondence Algorithm," in *European Conference on Computer Vision (ECCV)*, pp. 29–43.

Bhat, P., Curless, B., Cohen, M. F., and Zitnick, C. L. (2008), "Fourier Analysis of the 2D Screened Poisson Equation for Gradient Domain Problems," in *European Conference on Computer Vision (ECCV)*, pp. 114–128.

Brady, D. J., Gehm, M. E., Stack, R. A., Marks, D. L., Kittle, D. S., Golish, D. R., Vera, E. M., and Feller, S. D. (2012), "Multiscale gigapixel photography," *Nature*, 486, 386–389.

Brown, M. and Lowe, D. G. (2003), "Recognising panoramas," in *International Conference on Computer Vision (ICCV)*, vol. 2, pp. 1218–1225.

Brown, M., Szeliski, R., and Winder, S. (2005), "Multi-Image Matching Using Multi-Scale Oriented Patches," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 510–517.

Brox, T. and Malik, J. (2011), "Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33, 500–513.

Brox, T., Papenberg, N., and Weickert, J. (2004), "High Accuracy Optical Flow Estimation Based on a Theory for Warping," in *European Conference on Computer Vision (ECCV)*, vol. 4, pp. 25–36.

Burt, P. J., Edward, and Adelson, E. H. (1983), "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, 31, 532–540.

Cannon Inc. (2014), "Lens Calculator," http://www.canon.com/bctv/calculator/.

Darabi, S., Shechtman, E., Barnes, C., Goldman, D. B., and Sen, P. (2012), "Image Melding: Combining Inconsistent Images using Patch-based Synthesis," *ACM Transactions on Graphic*, 31, 82:1–82:10.

Debevec, P. E. and Malik, J. (1997), "Recovering high dynamic range radiance maps from photographs," in *SIGGRAPH*, pp. 369–378.

Efros, A. A. and Freeman, W. T. (2001), "Image quilting for texture synthesis and transfer," in *SIGGRAPH*, pp. 341–346.

Farbman, Z., Fattal, R., and Lischinski, D. (2011), "Convolution pyramids," *ACM Transactions on Graphic*, 30, 175:1–175:8.

Fattal, R., Lischinski, D., and Werman, M. (2002), "Gradient domain high dynamic range compression," *ACM Transactions on Graphic*, 21, 249–256.

Fischler, M. A. and Bolles, R. C. (1981), "RANSAC: Paradigm for Model," *Communications of the ACM*, 24.

F.Ray, S. (2002), *Applied Photographic Optics*, Focal Press.

Free Photography Tutorials, lessons, tips, tricks DSLR Camera (2014), "Camera Lenses," http://www.nobadfoto.com/lenses.html.

Freeman, M. (2008), *Mastering HDR Photography: Combining Technology and Artistry to Create High Dynamic Range Images*, Amphoto Books.

Gallo, O., Gelfand, N., Chen, W., Tico, M., and Pulli, K. (2009), "Artifact-free High Dynamic Range Imaging," in *IEEE Conference on Computational Photography*, pp. 1–7.

Golish, D. R., Vera, E. M., Kelly, K. J., Gong, Q., Jansen, P. A., Hughes, J. M., Kittle, D. S., Brady, D. J., and Gehm, M. E. (2012), "Development of a scalable image formation pipeline for multiscale gigapixel photography," *Optics Express*, 20, 22048–22062.

Granados, M., Ajdin, B., Wand, M., Theobalt, C., Seidel, H., and Lensch, H. (2010), "Optimal HDR reconstruction with linear digital cameras," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 215–222.

Grossberg, M. and Nayar, S. (2003), "Determining the Camera Response from Images: What is Knowable?" *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25, 1455–1467.

Gueguen, L., Pesaresi, M., and Soille, P. (2011), "An interactive image mining tool handling gigapixel images," in *International Geoscience and Remote Sensing Symposium*, IGARSS '11, pp. 1581–1584.

HaCohen, Y., Shechtman, E., Goldman, D. B., and Lischinski, D. (2011), "Non-Rigid Dense Correspondence with Applications for Image Enhancement," *ACM Transactions on Graphic*, 30, 70:1–70:9.

Hartley, R. I. and Zisserman, A. (2004), *Multiple View Geometry in Computer Vision*, Cambridge University Press, second edn.

Hasinoff, S. W., Durand, F., and Freeman, W. T. (2010), "Noise-Optimal Capture for High Dynamic Range Photography," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 553–560.

HDRsoft Ltd (2003), "Photomatix Pro," Software.

Heo, Y. S., Lee, K. M., Lee, S. U., Moon, Y., and Cha, J. (2011), "Ghost-free high dynamic range imaging," in *Asian conference on Computer Vision (ACCV)*, pp. 486–500.

Horisaki, R., Nakao, Y., Toyoda, T., Kagawa, K., Masaki, Y., and Tanida, J. (2009), "A thin and compact compound-eye imaging system incorporated with an image restoration considering color shift, brightness variation, and defocus," *Optical Review*, 16, 241–246.

Horn, B. K. and Schunck, B. G. (1981), "Determining optical flow," *Artificial Intelligence*, 17, 185–203.

Hu, J., Gallo, O., and Pulli, K. (2012), "Exposure Stacks of Live Scenes with Handheld Cameras," in *European Conference on Computer Vision (ECCV)*, pp. 499–512.

Hu, J., Gallo, O., Pulli, K., and Sun, X. (2013), "HDR Deghosting: How to deal with Saturation ?" in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1163 – 1170.

Iliopoulos, A.-S., Hu, J., Pitsianis, N., Sun, X., Gehm, M., and Brady, D. (2013), "Big Snapshot Stitching with Scarce Overlap," in *IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–6.

Jacobs, K., Loscos, C., and Ward, G. (2008), "Automatic High-Dynamic Range Image Generation for Dynamic Scenes," *IEEE Computer Graphics and Applications Magazine*, 28, 84–93.

Kang, S. B., Uyttendaele, M., Winder, S., and Szeliski, R. (2003), "High dynamic range video," *ACM Transactions on Graphic*, 22, 319–325.

Khan, E., Akyüz, A., and Reinhard, E. (2006), "Ghost Removal in High Dynamic Range Images," in *IEEE Computer Society Conference on Image Processing (ICIP)*, pp. 2005–2008.

Kim, S. J., Lin, H. T., Lu, Z., Süsstrunk, S., Lin, S., and Brown, M. S. (2012), "A New In-Camera Imaging Model for Color Computer Vision and its Application," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34, 2289–2302.

Kopf, J., Uyttendaele, M., Deussen, O., and Cohen, M. F. (2007), "Capturing and viewing gigapixel images," *ACM Transactions on Graphic*, 26.

Kopf, J., Kienzle, W., Drucker, S., and Kang, S. B. (2012), "Quality Prediction for Image Completion," *ACM Transactions on Graphic*, 31.

Levin, A., Zomet, A., Peleg, S., and Weiss, Y. (2006), "Seamless image stitching in the gradient domain," in *European Conference on Computer Vision (ECCV)*, vol. 4, pp. 377–389.

Liu, C. and Freeman, W. T. (2010), "A High-Quality Video Denoising Algorithm based on," in *European Conference on Computer Vision (ECCV)*, pp. 1–14.

Liu, C., Yuen, J., Torralba, A., Sivic, J., and Freeman, W. T. (2008), "SIFT Flow : Dense Correspondence across Different Scenes," in *European Conference on Computer Vision (ECCV)*, vol. 1, pp. 28–42.

London, B., Stone, J., and Upton, J. (2007), *Photography: The Essential Way*, Pearson.

Lucas, B. D. and Kanade, T. (1981), "An Iterative Image Registration Technique with an Application to Stereo Vision," in *Internationl Joint Conferences on Artifical Intelligence*, vol. 130, pp. 2480–2487.

Mann, S. (2000), "Comparametric equations with practical applications in quantigraphic image processing," *IEEE Transactions Image Processing (TIP)*, 9, 1389 –1406.

Mann, S. and Picard, R. (1995), "Being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures." in *IS&T 46th Annual Conference*, pp. 422–428.

Mantiuk, R., Myszkowski, K., and Seidel, H.-P. (2006), "A perceptual framework for contrast processing of high dynamic range images," *ACM Transactions Applied Perception*, 3, 286 – 308.

Matsushita, Y., Ofek, E., Tang, X., and yeung Shum, H. (2005), "Full-frame video stabilization," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 50–57.

McHugh, S. (2005), "Cambridge In Colour," http://www.cambridgeincolour.com.

Mertens, T., Kautz, J., and Reeth, F. V. (2007), "Exposure Fusion," in *Pacific Conference on Computer Graphics and Applications*, pp. 382–390.

Nayar, S. K. and Mitsunaga;, T. (2002), "Sptially Varying Pixel Exposures," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 472 – 479.

Nichols, M. H., Ruyle, G. B., and Nourbakhsh, I. R. (2009), "Very-high-resolution Panoramic Photography to Improve Conventional Rangeland Monitoring," *Rangeland Ecology & Management*, 62, 579–582.

Peleg, S., Ben-ezra, M., and Pritch, Y. (2001), "Omnistereo: Panoramic stereo imaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23, 279–290.

Pérez, P., Gangnet, M., and Blake, A. (2003), "Poisson image editing," *ACM Transactions on Graphic*, 22, 313–318.

Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., and Hoppe (2004), "Digital Photography with Flash and No-flash Image Pairs," *ACM Transactions on Graphic*, 23, 664–672.

Raman, S. and Chaudhuri, S. (2011), "Reconstruction of high contrast images for dynamic scenes," *Visual Computing*, 27, 1099–1114.

Reinhard, E., Heidrich, W., Debevec, P., Pattanaik, S., Ward, G., and Myszkowski, K. (2010), *High Dynamic Range Imaging, Second Edition: Acquisition, Display, and Image-Based Lighting*, Morgan Kaufmann.

Sen, P., Kalantari, N. K., Yaesoubi, M., Darabi, S., Goldman, D. B., and Shechtman, E. (2012), "Robust Patch-Based HDR Reconstruction of Dynamic Scenes," *ACM Transactions on Graphic*, 31, 203:1–203:11.

Shum, H.-y. and Szeliski, R. (1997), "Panoramic Image Mosaics Heung-Yeung Shum and Richard Szeliski," Tech. rep., Mircosoft Research.

Steinbr, F., Thomas, P., and Cremers, D. (2009), "Large Displacement Optical Flow Computation without Warping," in *International Conference on Computer Vision (ICCV)*, pp. 1609 – 1614.

Sun, D., Roth, S., and Black, M. J. (2010), "Secrets of Optical Flow Estimation and Their Principles," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2432 – 2439.

Szeliski, R. (2010), *Computer Vision: Algorithms and Applications*, Springer Verlag.

Thrun, S. (2011), "Google's dirverless car," .

Tico, M. and Pulli, K. (2009), "Image enhancement method via blur and noisy image fusion," in *IEEE Computer Society Conference on Image Processing (ICIP)*, pp. 1521 –1524.

Tomaszewska, A. and Mantiuk, R. (2007), "Image registration for multi-exposure high dynamic range image acquisition," in *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pp. 49–56.

Triggs, B., Mclauchlan, P., Hartley, R., and Fitzgibbon, A. (2000), "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*, pp. 298–375, Springer Verlag.

Tzimiropoulos, G., Argyriou, V., Zafeiriou, and Stathaki, T. (2010), "Robust FFT-Based Scale-invariant Image Registration with Image Gradients," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32, 1899–1906.

Uyttendaele, M., Eden, A., and Szeliski, R. (2001), "Eliminating Ghosting and Exposure Artifacts in Image Mosaics," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 509–516.

Venkataraman, K., Lelescu, D., Duparré, J., McMahon, A., Molina, G., Chatterjee, P., Mullis, R., and Nayar, S. (2013), "PiCam: An Ultra-Thin High Performance Monolithic Camera Array," *ACM Transactions on Graphic*, 32, 166:1–166:13.

Ward, G. (2003), "Fast, Robust Image Registration for Compositing high-dynamic Range Photographs from handheld exposures," *Journal of Graphics Tools*, 8, 17–30.

Wexler, Y., Shechtman, E., and Irani, M. (2007), "Space-Time Completion of Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29, 463–476.

Wilburn, B., Joshi, N., Vaish, V., Talvala, E.-V., Antunez, E., Barth, A., Adams, A., Horowitz, M., and Levoy, M. (2005), "High performance imaging using large camera arrays," *ACM Transactions on Graphic*, 24, 765–776.

Zhang, W. and Cham, W.-K. (2010), "Gradient-directed Composition of Multi-exposure Images," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 530–536.

Zhang, W. and Cham, W.-K. (2012), "Reference-guided exposure fusion in dynamic scenes," *Vision Coomunication and Image Representation*, 23, 467–475.

Zimmer, H. and Weickert, J. (2011), "Freehand HDR Imaging of Moving Scenes With Simultaneous Resolution Enhancement," *Computer Graphics Forum*, 30, 405–414.

Zitová, B. and Flusser, J. (2003), "Image registration methods: a survey," *Image and Vision Computing*, 21, 977–1000.

Biography

Jun Hu received B.Eng degree in Computer Science and Technology from Wuhan University, P.R.China, in 2009 and the M.Sci degree in Computer Science from Duke University in 2012. His research interests are in computational photography, numerical analysis and computer vision. At Duke University, he worked with Professor Xiaobai Sun on numerical analysis and its applications on computer vision. He worked at Nvidia research as Intern twice, in Fall 2011 and in Spring 2013. In Oct, 2013, he joined Apple Inc, where he is currently working on camera imaging algorithms for the next generation products.